ASYMPTOTIC AND FINITE-LENGTH OPTIMIZATION OF LDPC CODES

THÈSE N^{O} 3558 (2006)

PRÉSENTÉE LE 28 JUIN 2006

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Laboratoire de théorie des communications

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Abdelaziz AMRAOUI

ingénieur diplômé de l'Ecole Nationale de l'Aviation Civile, Toulouse, France
et de nationalité tunisienne

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury
Prof. R. Urbanke, directeur de thèse
Prof. H.-A. Loeliger, rapporteur
Dr A. Montanari, rapporteur
Prof. M. Shokrollahi, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2006

Abstract

This thesis addresses the problem of information transmission over noisy channels. In 1993, Berrou, Glavieux and Thitimajshima discovered Turbo-codes. These codes made it possible to achieve a very good performance at low decoding complexity. In hindsight, it was recognized that the underlying principle of using sparse graph codes in conjunction with message-passing decoding was the same as the one proposed in Gallager's remarkable thesis of 1963, although Gallager's work had all but been forgotten in the intervening 30 years. In 1997, there occurred a breakthrough in the analysis of this type of codes. Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann were able to give a complete characterization of the behavior of Low-Density Parity-Check code ensembles in the infinite blocklength case when used over the binary erasure channel. Soon thereafter, Richardson and Urbanke extended their results to binary-input memoryless symmetric channels.

In this thesis we present tools to analyze the performance of these types of codes and ways to optimize their parameters. The optimization for the infinite blocklength case is relatively straightforward and we give a simple and efficient method of doing so. However, this does not really solve the problem in practice, since the asymptotic analysis has only limited relevance for the short or moderate blocklengths that are typically used in practice. This brings us to the main objective of this thesis, which is to bridge the gap between the asymptotic case and the practical finite-length case. We follow the lead of Luby et al. by considering the binary erasure channel. We show that the performance of LDPC codes obeys a well-defined *scaling law* as the blocklength increases. This scaling law refines the asymptotic analysis and provides a good way to understand and approximate the behavior of LDPC codes of short to moderate length. We show how to compute the scaling parameters involved and demonstrate how to use the resulting approximation as a design and optimization tool.

Keywords: LDPC codes, low-density parity-check codes, factor graphs, error floor, density evolution, finite-length analysis, finite-length scaling, scaling law, analysis of LDPC codes, optimization of LDPC codes

Résumé

Cette thèse traite l'analyse des transmissions sur des canaux bruités. En 1993, Berrou, Glavieux et Thitimajshima découvrirent les Turbo-codes qui rendirent possible d'atteindre de très bonnes performances au prix d'une compléxité de décodage réduite. A posteriori, il fut reconnu que les principes sous-jacents au fonctionnement des Turbo-codes étaient identiques à ceux introduits par Gallager dans sa thèse en 1963. Ces principes, oubliés entre temps reposent sur l'utilisation de codes aux matrices de parité à faible densité en combinaison avec un décodage itératif se basant sur un échange de messages. En 1997, une percée dans l'analyse de ce type de codes permit à Luby, Mitzenmacher, Shokrollahi, Spielman et Stemann de caractériser de façon très précise le comportement des ensembles de codes "Low-Density Parity-Check" de longueur infinie et ce, lors de leur utilisation sur des canaux à effacement. Richardson et Urbanke étendirent ces résultats aux canaux à entrée binaire, sans mémoire et symétriques.

Dans cette thèse, nous présentons une analyse de la performance des codes LDPC. Nous commençons par étudier le cas des codes de longueur infinie, pour lequel nous exposons une méthode d'optimisation simple et efficace. Néanmoins, cela ne résout pas complètement le problème, puisqu'en pratique sont utilisés des codes courts ou de longueur moyenne. L'interêt des résultats obtenus au moyen de l'analyse asymptotique s'en retrouve réduit, ce qui nous amène à l'objectif principal de cette thèse qui est de faire le lien entre le cas asymptotique et le cas pratique de longueur finie. Nous suivons l'approche de Luby et al. en utilisant des canaux à effacement et nous montrons que la performance des codes LDPC suit une loi d'echelle bien définie lorsque la longueur augmente. Ceci nous permet d'affiner l'analyse asymptotique et de mieux comprendre le comportement des codes LDPC courts et de longueur moyenne. Nous expliquons comment calculer les paramètres régissant cette loi d'echelle et montrons comment l'approximation qui en découle peut être utilisée pour optimiser les performance des codes LDPC.

Mots Clés: Codes LDPC, codes à matrice de parité de faible densité, graphes de facteurs, palier d'erreur, évolution de densité, analyse en longueur finie, loi d'echelle, analyse des codes LDPC, optimisation des codes LDPC.

Acknowledgments

I am deeply indebted to my advisor, Professor Rüdiger Urbanke, for his constant support. Without his commitment and dedication, this work would not have been possible. I want to thank him for the chance that I had to learn so much from him, through the long hours that we spent working together and the constant interactions that we had.

I would like to thank Dr. Andrea Montanari for his insightful suggestions and his help that contributed greatly to this work. I would also like to thank the other members of my committee Prof. Emre Telatar, Prof. Hans-Andrea Loeliger and Prof. Amin Shokrollahi.

During these years at EPFL, I had the chance to interact with so many interesting people from which I learned so much especially the other Professors that headed our group, Prof. Emre Telatar, Prof. Bixio Rimoldi and Prof. Suhas Diggavi. I also want to thank them for providing us with such nice work environment and guidance.

I would also like to thank Dr. Gerhard Kramer and Prof. Shlomo Shamai for their supervision during my visit to Bell Labs.

On a more personal level, thanks to the people that were around, my time at EPFL has been both interesting and exciting. I would like to thank all the members of our labs that I consider as friends, in their order of appearance, Rüdiger, Emre, Bixio, Muriel, Denice, Aslan, Cyril, Changyan, Olivier, Sanket, Shashi, Sibi, Nicolae, Ninoslav, Rajesh, Prasenjit, Linus, Stephane, Tarek, Daniella, Vish, Peter, Jérémie, Suhas, Christina, Etienne, Nicolas, Christine, Dinkar, Shrinivas, John, Henry, Satish, Voja, Jasper, Marius, Ayfer, Harm...

Finally, I would like to thank my family for their constant support and encouragement during this long endeavor.

To my parents...

Contents

Chapter 1

Introduction

The subject of this thesis is the analysis of Low-Density Parity-Check (LDPC) codes when decoded using Belief Propagation (BP). Our aim is to contribute to the understanding of how these decoding schemes function and to provide analysis tools that can be used in their design and optimization. The approach we follow is to study how the performance of LDPC codes scales with increasing blocklengths. We show empirically that the scaling law which we prove for the binary erasure channel (BEC) gives us an accurate assessment of the performance already for moderate and short blocklengths. This enables us to perform a finite-length optimization.

Although this does not completely solve the question of how to optimally choose finite-length codes (since the standard irregular ensembles which we consider are simply not powerful enough and better classes of ensembles exist), this approach is in principle applicable to a wide range of cases.

LDPC codes were originally defined and analyzed by Gallager in his 1963 Ph.D. thesis [1]. They were then long forgotten except for a few interesting contributions by Pinsker and Zyablov [2], Tanner [3], and later Wiberg [4] and MacKay [5]. The discovery of Turbo-Codes in 1993 by Berrou, Glavieux and Thitimajshima [6] showed the potential of iterative decoding. It was not immediately clear, however, how to explain the record-breaking performance that was observed in practice.

The first breakthrough in the analysis of iterative codes came in a series of papers by Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann [7, 8, 9, 10]. In these papers, several new key concepts were introduced. The first idea was to concentrate on the BEC. The nature of

this channel enabled Luby et al. in [7] to use tools from discrete stochastic processes and graph theory in their analysis. The second concept was to consider ensembles of codes rather than specific instances. In this respect, Luby et al. showed that the performance of specific instances concentrates around the ensemble average, so that this average constitutes a relevant quantity. The main result of their analysis is that they were able to predict the asymptotic (in the blocklength) performance of LDPC codes and to find capacity achieving sequences of degree distributions for the BEC.

Richardson and Urbanke [11] extended several of these concepts to a more general class of channels, showing that the main conclusions drawn while studying the BEC have counterparts for all binary-input memoryless symmetric channels. The generalized version of density evolution was used in [12, 11, 13] to find LDPC code ensembles that can, asymptotically in the blocklength, operate at $0.0045dB$ away from capacity on an additive white Gaussian noise channel. Using the convenient and unifying framework of factor graphs developed by Wiberg, Loeliger, Kötter in [14] and Kschischang, Frey, and Loeliger in [15], these new tools were harnessed to devise and analyze numerous communication schemes and strategies for a wide range of applications and using different kinds of codes (Turbo-Codes, IRA, Rateless,...).

Unfortunately, the convergence of the behavior of LDPC codes or other families to the asymptotic limits is quite slow and therefore, results drawn from the asymptotic analysis (density evolution) are of limited relevance if one considers short or moderate blocklengths. This was the reason why Di, Proietti, Richardson, Telatar, and Urbanke [16] directly analyzed the finite-length behavior of LDPC code ensembles on the BEC through the definition of stopping sets and the analysis of their distribution during the decoding process. Their combinatorial approach unfortunately still has some shortcomings. The problem is that it is computationally quite costly to implement, which prevents us from using it as a design and optimization tool. Furthermore, it is not clear that this approach can be extended to a wider class of channels.

In this thesis, we analyze the behavior of LDPC codes when used over the BEC for moderate and short blocklengths. Our main contribution is that, under some technical conditions, the block and bit error probabilities behave like

$$\mathrm{P}_{\mathrm{B},\gamma}(n,\lambda,\rho,\epsilon) = Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right)(1+o(1))\,,$$

$$\mathrm{P}_{\mathrm{b},\gamma}(n,\lambda,\rho,\epsilon) = \nu^* Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right)(1+o(1))\,,$$

where the limit is taken such that $\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)$ is kept constant while the blocklength n increases. The parameters $\epsilon^* = \epsilon^*(\lambda, \rho)$, $\nu^* = \nu^*(\lambda, \rho)$, $\alpha = \alpha(\lambda, \rho)$ and $\beta = \beta(\lambda, \rho)$ are constants which depend on the degree distributions defining the ensemble and the Q-function is defined as usual $Q(x) = \int_x^{+\infty} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx$.

We will show that this result can be used to obtain a good approximation of the error probability curves and also to optimize the degree distributions. The existence of such a scaling law in the context of channel coding was first suggested by Montanari in [17].

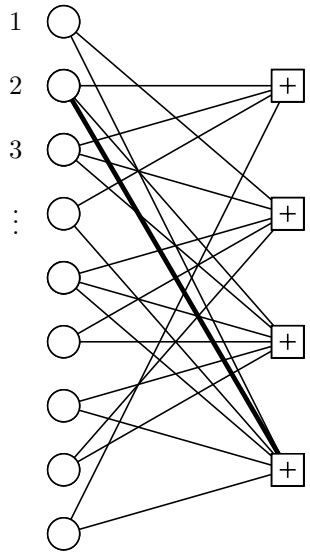
In the remainder of this chapter, we will describe the ensembles of LDPC codes and the decoding algorithm we consider as well as the notation associated to them. We will also introduce our approach to the analysis of the performance of LDPC codes and describe the structure of the thesis.

1.1 Low-Density Parity-Check Code Ensembles

Low-Density Parity-Check codes are binary linear block-codes that have a sparse parity-check matrix. Tanner introduced in [3] a convenient graphical representation of LDPC codes in terms of a bipartite graph. This representation is probably best described as the factor graph [14, 15] associated to the BP decoding of LDPC codes. It is useful in the encoding [18], decoding, as well as in the analysis of LDPC codes.

To find the bipartite graph associated to a code, represent each column of the parity-check matrix or equivalently each bit of the codeword by a variable node depicted by a circle. Represent each row of the parity-check matrix or equivalently each parity-check relation by a check node depicted by a square. Draw an edge between a variable and a check node if there is a one in the intersection of the corresponding column and row of the parity-check matrix or, in other words, if the corresponding bit is involved in the parity-check relation. The degree of a node in the graph is the number of edges emanating from that node. The variable degree of an edge is the degree of the variable node that edge is connected to and similarly the check degree of an edge is the degree of the check node the edge is connected to. An example is shown in Fig.1.1.

In this thesis, we consider ensembles of LDPC codes defined directly through their bipartite graphs by three main parameters. The blocklength n, the edge perspective variable node distribution $\lambda(x) = \sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i x^{i-1}$ and finally the edge perspective check node degree distribution $\rho(x) = \sum_{i=1}^{\mathtt{r}_{\max}} \rho_i x^{i-1}$. To pick an element from the ensemble define n variable nodes and



$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & \mathbf{1} & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Figure 1.1: A parity-check matrix H and its corresponding bipartite graph. The thick edge connects the second variable node to the last check node corresponding to the 1 at the intersection of the corresponding column and row.

$n_c = n\frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$ check nodes. Assign the degrees of the nodes such that a fraction λ_i of the edges is connected to variable nodes of degree $i \leq \mathtt{l}_{\max}$, with $\mathtt{l}_{\max}$ the maximum variable, or left, degree and that a fraction ρ_i of the edges is connected to check nodes of degree $i \leq \mathtt{r}_{\max}$, with $\mathtt{r}_{\max}$ the maximum check, or right, degree. There is the same number ξ of edges emanating from the variable nodes on one side and from the check nodes on the other. Label the edges emanating from the variable nodes from 1 to ξ and similarly the edges emanating from the check nodes. Generate a permutation uniformly at random from the set of all permutations of length ξ and finally, connect the edges emanating from the variable nodes to the edges emanating from the check nodes according to this permutation. We denote this ensemble by LDPC$(n, \lambda(x), \rho(x))$.

Apart form the edge perspective degree distributions, we define the node perspective degree distributions $\Lambda(x) = \sum_{i=1}^{\mathtt{l}_{\max}} \Lambda_i x^i = \frac{\int_0^x \lambda(u)du}{\int_0^1 \lambda(u)du}$ for the variable nodes and $P(x) = \sum_{i=1}^{\mathtt{r}_{\max}} P_i x^i = \frac{\int_0^x \rho(u)du}{\int_0^1 \rho(u)du}$ for the check nodes. Λ_i is the fraction of variable nodes of degree i in the graph, for $i \leq \mathtt{l}_{\max}$ and similarly P_i is the fraction of check nodes of degree i the graph, for $i \leq \mathtt{r}_{\max}$. Finally, the rate of the code can be computed from the degree distribution in the following way, $rate = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = 1 - \frac{\sum_{i=1}^{\mathtt{r}_{\max}} \frac{\rho_i}{i}}{\sum_{j=1}^{\mathtt{l}_{\max}} \frac{\lambda_j}{j}}$.

1.2 Belief Propagation Decoding

Assume transmission occurs over a binary-input symmetric memoryless channel. We denote the transmitted word of length n by X and its n components by $X_i \in \{-1, 1\}, 1 \leq i \leq n$. The decoder receives the vector Y with components Y_i, $1 \leq i \leq n$. The channel is memoryless meaning that the transition probability can be written as $p(Y|X) = \prod_{i=1}^n p(Y_i|X_i)$, and symmetric so that $p(Y_i|X_i = 1) = p(-Y_i|X_i = -1)$, $i \in \{1, \cdots, n\}$.

Consider now the BP decoding process. The decoder uses the bipartite graph as a support and proceeds by exchanging messages on the edges of the graph. Each edge carries two messages, one sent from the variable to the check node and one from the check to the variable node. The BP algorithm runs as described in Alg. 1.

Intuitively, BP attempts to compute the log-likelihood of each bit. It starts with a log-likelihood for each bit x_i that involves only the observation y_i and keeps increasing the set of observations it takes into account at each iteration. It is easy to show that BP computes exactly the log-likelihood ratios of each bit x_i based on the whole received vector on a loop-free graph. However, whenever in a bipartite graph, the number of iterations exceeds the girth of the graph

Algorithm 1 Belief Propagation

Initialization:

Decoder receives the vector y.

Initialize all messages in graph to 0.

Assign to each variable node the log-likelihood ratio $\mu_i = \log\left(\frac{p(y_i|x_i=0)}{p(y_i|x_i=1)}\right)$, $1 \le i \le n$.

Iterations:

for $k = 0$ to l **do**

Compute all messages sent out from variable nodes.

for $i = 1$ to n **do**

Consider variable node i of degree d.

The incoming messages are $\mu_{in}^1, \cdots, \mu_{in}^d$.

The j^{th} outgoing message ($1 \le j \le d$) is computed with the following rule

$$\mu_{out}^j = \mu_i + \sum_{m=1, m\neq j}^{d} \mu_{in}^m$$

end for

Compute all messages sent out from check nodes.

for $i = 1$ to n_c **do**

Consider check node i of degree d.

The incoming messages are $\mu_{in}^1, \cdots, \mu_{in}^d$.

The j^{th} outgoing message ($1 \le j \le d$) is computed with the following rule

$$\mu_{out}^j = 2 \operatorname{arctanh}\left(\prod_{m=1, m\neq j}^{d} \tanh(\mu_{in}^m/2)\right)$$

end for

end for

Decisions:

for $i = 1$ to n **do**

Consider variable node i of degree d.

if $\left(\mu_i + \sum_{m=1}^{d} \mu_{in}^m\right) > 0$ **then**

decision$(i) = 0$

end if

if $\left(\mu_i + \sum_{m=1}^{d} \mu_{in}^m\right) < 0$ **then**

decision$(i) = 1$

end if

if $\left(\mu_i + \sum_{m=1}^{d} \mu_{in}^m\right) = 0$ **then**

decision$(i) = 0$ or 1 with probability one half.

end if

end for

the values computed by the algorithm are no longer exact log-likelihood ratios, which makes BP suboptimal.

In the case of the BEC, several simplifications arise. The initial log-likelihood are either 0 (erased message) or $\pm\infty$ (known message) and one can show that throughout the decoding all messages exchanged in the graph remain of this type. The decoding rules at each node can therefore be expressed in a simpler form. A variable node sends a "known message" on an edge if it receives at least one "known message" on its other edges and a check node sends a "known message" on an edge only if all the other messages it receives are "known".

It was shown in [16] that on the BEC, if the BP decoder is not successful, then the remaining subset of variable nodes forms a stopping set. A stopping set $\mathcal{S}$ is a set of variable nodes such that all check nodes which are connected to $\mathcal{S}$ are connected at least twice. To see that this is true, consider the following. If all variable nodes composing a stopping set are erased by the channel, then all check nodes connected to the stopping set (and that could potentially correct the nodes) receive at least two erased messages and therefore send erased messages on all their edges. For this reason, the set of variable nodes that remain undecoded after BP is equal to the largest stopping set contained in the erased set. If the erased set of variable nodes does not contain any stopping set, then the decoder is successful.

In [7], the Peeling Algorithm (PA) for the BEC was introduced. This algorithm that we describe in Alg. 2 works on the bipartite graph by correcting the variable nodes in a greedy fashion and removing the corrected nodes from the graph. In more detail, upon receiving the channel output, the known variable nodes send their values to the check nodes and are removed from the graph. The decoder proceeds by looking for a check node of degree one, meaning that all but one bit in the corresponding parity-check relation are known. If it finds one, it corrects the value of the variable connected to it, propagates this information to all check nodes connected to the variable and removes it from the graph before proceeding. If it does not find one, then the decoding stops. The residual graph has at that point no degree-one check nodes, meaning that all check nodes are connected at least twice. Therefore, the residual graph forms a stopping set. Furthermore, we have that the PA leaves undecoded the largest stopping set in the erased set. To see that this is true, consider the case where all variable nodes composing a stopping set are erased by the channel. The check nodes connected to them, will always remain of degree higher or equal to two during the decoding process and will therefore not be able to decode any variable in the stopping set.

Algorithm 2 Peeling

Initialization: $t = 0$

Decoder receives the vector y.

Variables send received values to check nodes that keep memory of them.

Known variable nodes and all edges connected to them are removed from graph.

Iterations:

while true **do**

  if there is a check node of degree one in the residual **then**

    Send its value to the variable node connected to it.

    This variable sends its value to all checks it is connected to and is removed with its edges.

    $t = t + 1$.

  else

    break

  end if

end while

Decisions:

if residual graph is empty **then**

  decoding is successful.

else

  the residual graph is a stopping set.

end if

For this reason, BP and PA are completely equivalent in terms of their performance. However, the PA lends itself more readily to analysis. We will explain in more detail this analysis in Section 2.2.

1.3 Motivation and Outline

Consider the ensemble LDPC$(n, \lambda(x), \rho(x))$ and transmission over a binary-input memoryless symmetric channel. This can be a BEC that erases the transmitted bits with probability ϵ or an additive white Gaussian noise channel that adds a zero mean Gaussian noise of standard deviation σ to the transmitted BPSK symbols.

How can one analyze the performance of such a transmission?
The simplest solution would be to simulate and observe the error probability. This, however provides little insight into why the performance is as observed and provides no or few clues on how to improve the code. For infinite blocklengths and ensembles that have a vanishing error floor, density evolution enables us to compute the threshold of the code (see [7, 11]) from the degree distributions $\lambda(x)$ and $\rho(x)$. For channel parameters (ϵ for the BEC) below the threshold, error-free transmission is asymptotically possible and above it is not. Therefore, density evolution is useful in predicting the performance of LDPC codes for large blocklengths, and in this setting it can be used to optimize degree distributions. In Section 4.1 we will discuss how to optimize thresholds efficiently.

However, as one can see in Fig. 1.2, the threshold of the code gives little information on the performance for short or moderate blocklengths. Consider the ensemble LDPC$(n = 4096, \lambda(x) = x^2, \rho(x) = x^5)$ and transmission over a BEC with $\epsilon = 0.41$. Density evolution tells us that if we are transmitting below the threshold then the error probability should vanish with increasing blocklength. Nevertheless, we see in Fig. 1.2 that the block error probability for $n = 4096$ is still $P_B = 0.024455$.

Di et al. showed in [16] how to compute the average error probability curves exactly for regular low-density parity-check code ensembles of any length when used over the BEC. They showed that in this case, the finite-length analysis boils down to a combinatorial problem. The recursions they provide were generalized in [19] to deal with irregular ensembles, expurgation and to compute block as well as bit error probabilities. However, in practice their approach runs into computational limitations. The complexity of the recursions grows by a factor n for each

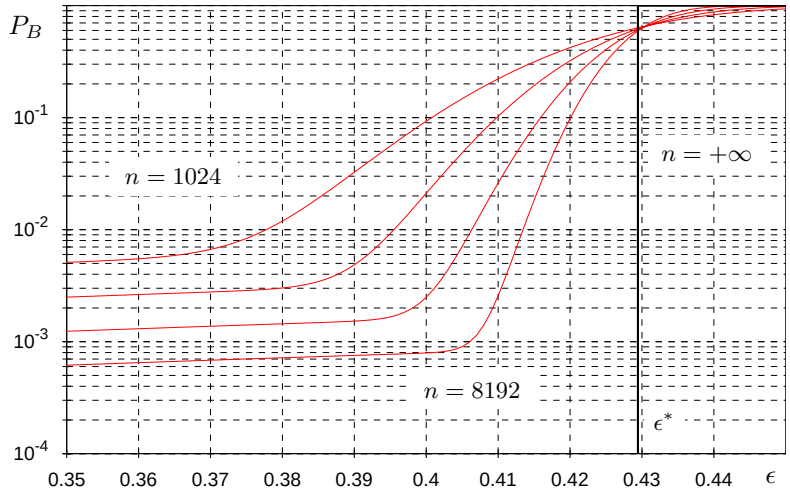


Figure 1.2: Block error probability curves for LDPC$(n, \lambda(x) = x^2, \rho(x) = x^5)$ when used over a BEC of erasure probability ϵ. The different curves are for $n \in \{1024, 2048, 4096, 8192\}$. The threshold is $\epsilon^* = 0.42943981$

additional degree of nodes considered in the ensemble. Therefore, even for short blocklengths, only very simple ensembles can currently be analyzed in this way.

It is therefore of great interest to find an alternative analysis that can be used to predict the performance of LDPC codes in practical cases and as well as for their design. This is the main objective of the thesis.

We address this issue in the following way. There is a clear distinction between two parts in the curves in Fig. 1.2. For small channel erasure probabilities, the curve is flat. This corresponds to the error floor and is due to small (sublinear in the blocklength) erased stopping sets remaining after the decoding. The asymptotic distribution of stopping sets has already been studied in [20] and we will show in Section 3 how to construct a good approximation for the error probability curves in this region. For larger erasure probabilities, close to the threshold the curves become steeper. This is called the waterfall region and is due to large (linear in the blocklength) failures in the decoding. Much less is known about the behavior of the codes in this region.

In this thesis, we will show that for the BEC, the error probability curves around the threshold are governed by a well defined scaling law ([17, 21, 22, 23]). In Section 2 we will prove this scaling law and show how it can be used to find an accurate approximation of the error probability curves in the waterfall region. Combining this approximation with the one in the error floor region will

enable us to accurately predict the performance of the codes (see Fig. 1.3). Finally, we will show

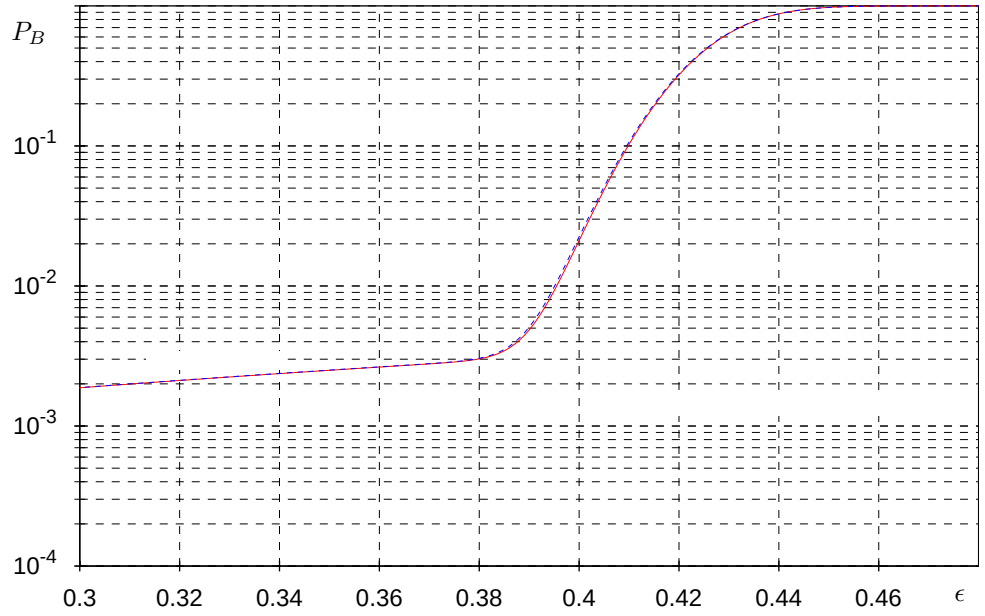


Figure 1.3: Block error probability curves for LDPC($n = 2048, \lambda(x) = x^2, \rho(x) = x^5$) when used over a binary erasure channel of erasure probability ϵ. The solid curve is the exact error probability curve and the dashed curve is obtained with our approximation.

how this approximation can be used to optimize and find the best degree distributions for a particular blocklength, and channel erasure probability in Section 4.2.

Chapter 2

Waterfall

2.1 Motivation

In this chapter, we address the issue of modeling the behavior of large error events in the decoding process. Our approach is motivated by a general conjecture stemming from statistical physics [21, 22]: If a system, parametrized by some variable ϵ, goes through a *phase transition* at a critical parameter, call it ϵ^* (in our case the threshold), then it has repeatedly been observed that around this critical parameter there is a very specific scaling law. To be more concrete: We are interested in the probability of block error as a function of the blocklength n and the channel parameter ϵ, call it $\mathrm{P_B}(n, \epsilon)$. We know that as n tends to infinity there is a phase transition at ϵ^*, the iterative decoding threshold. Asymptotically, $\mathrm{P_B}(n, \epsilon)$ tends to zero for $\epsilon < \epsilon^*$ and to one for $\epsilon > \epsilon^*$. The scaling law refines this basic observation: One expects that there exists a non-negative constant μ and some non-negative function $f(z)$ so that

$$\lim_{\substack{n \to \infty \\ \text{s.t. } n^{1/\mu}(\epsilon^* - \epsilon) = z}} \mathrm{P_B}(n, \epsilon) = f(z). \tag{2.1}$$

In other words, if one plots $\mathrm{P_B}(n, \epsilon)$ as a function of $z = n^{\frac{1}{\mu}}(\epsilon^* - \epsilon)$ then, for increasing n these finite-length curves are expected to converge to some function $f(z)$. The function $f(z)$ decreases smoothly from 1 to 0 as its argument changes from $-\infty$ to $+\infty$. This means that all finite-length curves are, to first order, scaled versions of some *mother* curve $f(z)$. It might be helpful to think of the threshold ϵ^* as the zero order term in a Taylor series. Then the above scaling, if correct, represents the first order term. In fact, one can even refine the analysis to

include higher order terms and write

$$\mathrm{P_B}(n,\epsilon) \quad = \quad f(z) + n^{-\omega} g(z) + o(n^{-\omega}),$$

where ω is some positive real number and $g(z)$ is the second order correction term.

We will show in this chapter that this is indeed the case for the error probability of LDPC code ensembles when used over the BEC. Let us start by reviewing how the finite and asymptotic blocklength decoding of LDPC codes can be studied over the BEC.

2.2 Decoding Analysis

Consider the ensemble LDPC$(n, \lambda(x), \rho(x))$, transmission over the BEC and decoding using the PA. At any step of the decoding process parametrized by the discrete time $t \in \mathbb{N}$, we define the state vector of the decoder to be $X_t = (\mathrm{R}_1, \mathrm{R}_2, \cdots, \mathrm{R}_{\mathtt{r}_{\max}-1}, \mathrm{L}_1, \mathrm{L}_2, \cdots, \mathrm{L}_{\mathtt{l}_{\max}})$ with L_i, $1 \leq i \leq \mathtt{l}_{\max}$, being the number of edges connected to degree i variable nodes in the residual graph and R_i, $1 \leq i \leq \mathtt{r}_{\max}$, being the corresponding quantity for the check nodes. $\mathrm{R}_{\mathtt{r}_{\max}}$ is not present in the state vector as it can be computed from the other components: using the relation $\sum_{i=1}^{\mathtt{l}_{\max}} \mathrm{L}_i = \sum_{i=1}^{\mathtt{r}_{\max}} \mathrm{R}_i$, we obtain $\mathrm{R}_{\mathtt{r}_{\max}} = \sum_{i=1}^{\mathtt{l}_{\max}} \mathrm{L}_i - \sum_{i=1}^{\mathtt{r}_{\max}-1} \mathrm{R}_i$.

From the construction of the ensemble LDPC$(n, \lambda(x), \rho(x))$ and the characteristics of the PA it is clear that the vector X_t is indeed a valid state for our decoding process and that furthermore, the process is a first-order Markov process. In other words, the transition probabilities from one state to another during a decoding round are independent of which particular instance is involved and depend only on the current state. Furthermore, knowing the evolution of this state vector during the decoding is sufficient to characterize its success or failure. As explained in Alg. 2, the decoder stops if the coefficient of the vector $\mathrm{R}_1 = 0$ before the residual graph is empty. The transition probabilities of this Markov process have already been presented in [7] and will be used in Section 2.4.2.

If one is able to compute the exact distribution of this Markov process throughout the decoding, one obtains the block and bit error probabilities. For example, to find the block error probability, it would suffice to consider the distribution of the Markov process at the end of the decoding and sum the probabilities of all states compatible with $\mathrm{R}_1 = 0$ and a non-empty residual graph. The analysis of Luby et al. in [7, 8] consisted of computing the expected states of this Markov process for asymptotic blocklengths and in showing that individual realizations concentrate around this mean for increasing blocklengths.

Let us recall the results of [7, 8]. It was shown that at any step $t \in \mathbb{N}$ of the decoding one can write that almost surely

$$\mathrm{L}_i = \xi l_i(t/\xi) + O(\xi^{5/6}), \qquad i \in \{1, \cdots, \mathtt{l}_{\max}\} \tag{2.2}$$

$$\mathrm{R}_i = \xi r_i(t/\xi) + O(\xi^{5/6}), \qquad i \in \{1, \cdots, \mathtt{r}_{\max}\} \tag{2.3}$$

where $\xi = n\Lambda'(1)$ is the total number of edges in the graph and $\tau = t/\xi$ is the "normalized time". A more convenient parametrization is the following. Define the variable y such that $d\tau/dy = -\epsilon\lambda(y)$ and $y = 1$ when $\tau = 0$. Then if we parametrize our decoding by y, we can write

$$l_i(y) = \epsilon\lambda_i y^i, \qquad i \in \{1, \cdots, \mathtt{l}_{\max}\} \tag{2.4}$$

$$r_1(y) = \epsilon\lambda(y)[y - 1 + \rho(1 - \epsilon\lambda(y))] \tag{2.5}$$

$$r_i(y) = \sum_{m \geq j \geq i} (-1)^{i+j} \binom{j-1}{i-1} \binom{m-1}{j-1} \rho_m(\epsilon\lambda(y))^j \qquad i \in \{2, \cdots, \mathtt{r}_{\max}\}. \tag{2.6}$$

Studying the function $r_1(y)$ for $0 \leq y \leq 1$ tells us whether the decoder will be asymptotically successful or not and provides us with the threshold of the code ϵ^*:

- If $\epsilon < \epsilon^*$, then $r_1(y) > 0$ for all $y \in (0, 1]$. This means that the decoder will be successful asymptotically with high probability.
- If $\epsilon > \epsilon^*$, then there exists a $y \in (0, 1]$ such that $r_1(y) = 0$. This means that the decoder will fail asymptotically with high probability.

This is shown in Fig. 2.1 where we see that for ϵ^*, the curve of $r_1(y)$ touches the zero line. We call a point where $r_1(y)$ touches the zero line and has zero derivative a *critical point*. A code can have several critical points. Those can correspond to the same or to different channel parameters ϵ. Critical points associated to the lowest channel parameter determine the threshold.

We already know that the BP and the PA fail on the same set of variable nodes, which corresponds to the maximal stopping set contained in the erased set. In fact, their behavior is related not only at the end of the decoding process but actually throughout. In order to see this relationship, think of the following experiment. Consider the BP decoder at the ℓ-th iteration for an infinite graph. Call y_ℓ, the fraction of erased messages that are sent from the check nodes to the variable nodes during this iteration. Now, take the decisions on all variable nodes and remove those that are known with their edges from the graph. What is the degree distribution of the residual graph obtained in this way? Interestingly, computing this degree distribution

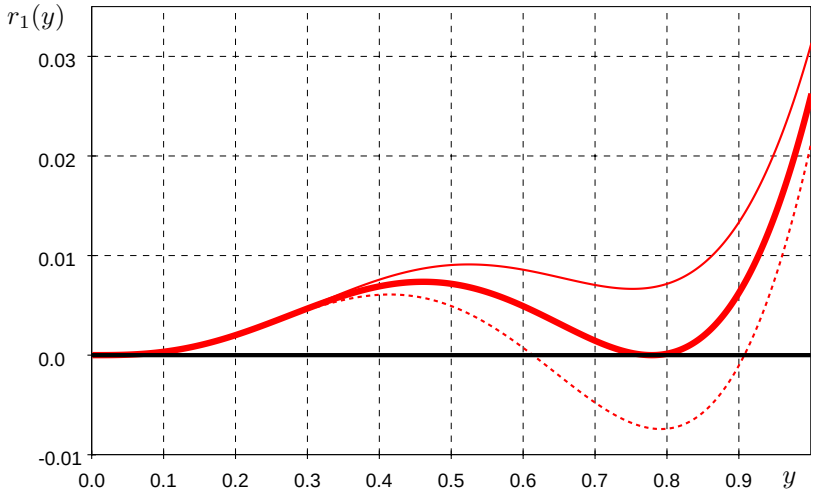


Figure 2.1: The asymptotic fraction $r_1(y)$ for $\lambda(x) = x^2$ and $\rho(x) = x^5$. The thick curve is for $\epsilon* = 0.4294381$. The solid curve is for $\epsilon = 0.4 < \epsilon^*$ and the dashed curve is for $\epsilon = 0.46 > \epsilon^*$.

leads to exactly the same expressions as given in (2.4)-(2.6). One can therefore think of y that parametrizes the solution of the PA as the fraction of erased messages that are sent from check nodes to variable nodes in the BP decoder. One can show that if this quantity is y at the ℓ-th iteration of BP, it becomes $1 - \rho(1 - \epsilon\lambda(y_\ell))$ in the $(\ell + 1)$-th iteration while it varies in a continuous way between the two values in the PA counterpart. Characterizing the success or failure of the BP algorithm becomes equivalent to analyzing the evolution of y during the iterations. It converges either to zero or to another fixed point. In the first case, the decoder is successful and otherwise the characteristics of the remaining residual can be found by plugging the fixed point y into (2.4)-(2.6). Similarly, one can observe the evolution of the fraction of erased messages sent from variable to check nodes in the BP decoder denoted by x. This analysis is called *density evolution* and presented in much more detail in [20].

In the rest of this chapter and in the following, we will use the resulting set of *critical* values $(\epsilon^*, \tau^*, y^*, x^*, \nu^*)$:

- ϵ^*, the threshold of the code
- τ^*, the asymptotic critical time, t is the root of $r_1(\tau)$ (parametrized by the normalized time) at the threshold
- y^*, the asymptotic fraction of erased messages sent from check to variable nodes, it is the

root of $r_1(y)$ at the threshold

- x^*, the asymptotic fraction of erased messages sent from variable to check nodes at the threshold, it is equal to $x^* = \epsilon^* \lambda(y^*)$
- ν^*, the asymptotic fraction of undecoded variable nodes at the threshold, it is equal to $\nu^* = \epsilon^* \Lambda(y^*)$

As we have already seen (see Fig. 1.2), the threshold obtained through density evolution is not sufficient to predict the performance of LDPC codes of moderate or short blocklengths. In the following sections, we will show and use the fact that the distribution of the state vectors converges weakly to a Gaussian and we provide means to compute its mean and covariance matrix through a set of differential equations that we term *covariance evolution*. This will be done in Lemma 4. In this lemma, we use a generic form for a finite-dimensional Markov process satisfying certain properties and later apply it to our decoding process.

2.3 Finite-Length Scaling

2.3.1 The Scaling Law

The main result of the thesis is described in the following lemma.

Lemma 1. [Basic Scaling Law] Consider transmission over a BEC of erasure probability ϵ using random elements from an ensemble LDPC(n, λ, ρ) which has a single non-zero critical point at the threshold $\epsilon^* = \epsilon^*(\lambda, \rho)$ and denote by ν^* the asymptotic fractional size of the residual graph at this critical point. Fix z to be $z := \sqrt{n}(\epsilon^* - \epsilon)$. Let $\mathrm{P}_{\mathrm{b},\gamma}(n, \lambda, \rho, \epsilon)$ denote the expected bit error probability and let $\mathrm{P}_{\mathrm{B},\gamma}(n, \lambda, \rho, \epsilon)$ denote the expected block error probability *due to errors of size at least* $\gamma\nu^*$, where $\gamma \in (0, 1)$. Then as n tends to infinity,

$$\mathrm{P}_{\mathrm{B},\gamma}(n, \lambda, \rho, \epsilon) = Q\left(\frac{z}{\alpha}\right)(1 + o_n(1)),$$
$$\mathrm{P}_{\mathrm{b},\gamma}(n, \lambda, \rho, \epsilon) = \nu^* Q\left(\frac{z}{\alpha}\right)(1 + o_n(1)),$$

where $\alpha = \alpha(\lambda, \rho)$ is a constant which depends on the ensemble.

We conjecture that in fact the following refined scaling law is valid.

Conjecture 1. [Refined Scaling Law] Consider transmission over a BEC of erasure probability ϵ using random elements from an ensemble LDPC(n, λ, ρ) which has a single non-zero critical point

at the threshold $\epsilon^* = \epsilon^*(\lambda, \rho)$ and denote by ν^* the asymptotic fractional size of the residual graph at this critical point. Fix z to be $z := \sqrt{n}(\epsilon^* - \epsilon)$. Let $\mathrm{P}_{\mathrm{b},\gamma}(n, \lambda, \rho, \epsilon)$ denote the expected bit error probability and let $\mathrm{P}_{\mathrm{B},\gamma}(n, \lambda, \rho, \epsilon)$ denote the expected block error probability *due to errors of size at least* $\gamma\nu^*$, where $\gamma \in (0, 1)$. Fix z to be $z := \sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)$. Then as n tends to infinity,

$$\mathrm{P}_{\mathrm{B},\gamma}(n, \lambda, \rho, \epsilon) = Q\left(\frac{z}{\alpha}\right)\left(1 + O(n^{-1/3}\right),$$
$$\mathrm{P}_{\mathrm{b},\gamma}(n, \lambda, \rho, \epsilon) = \nu^* Q\left(\frac{z}{\alpha}\right)\left(1 + O(n^{-1/3}\right),$$

where $\alpha = \alpha(\lambda, \rho)$ and $\beta = \beta(\lambda, \rho)$ are constants which depend on the ensemble.

Note that this scaling does not apply to codes whose threshold is determined by the stability condition (in that case, zero is a critical point at the threshold). Those ensembles exhibit a very different scaling law which is described in Appendix A.

We prove Lemma 1. in Section 2.4. Our approach is to consider first a situation slightly simplified with respect to the one encountered in iterative decoding. We later show that the main conclusions hold true when the simplifying assumptions are removed. Conjecture 1 will be proven in the simplified context and we provide some heuristic argument suggesting that the simplifying assumptions are in fact irrelevant. But to start we will show how the scaling parameters α and β are computed before giving an informal justification of the above scaling laws in Section 2.3.3.

2.3.2 The Scaling Parameters

The Scaling Parameter α

The value of scaling parameter α that describes the slope of the error probability curves is given by the following Lemma.

Lemma 2. [Scaling Parameter α] Consider transmission over a BEC using random elements from an ensemble LDPC$(n, \lambda(x), \rho(x))$. The scaling parameter α in Lemma 1 is given by

$$\alpha = \frac{\sqrt{\delta^{r_1, r_1}(y^*)}}{\sqrt{\Lambda'(1)}\epsilon^* \lambda(y^*)^2 \rho'(1 - \epsilon^* \lambda(y^*))},$$

where ϵ^* is the threshold of the ensemble and y^*, the non-zero solution of $r_1(y^*) = 0$, is the critical value defined in Section 2.2. Further, $\delta^{r_1, r_1}(y^*) = \delta^{00}(y^*)$ is the normalized variance of

the number of check nodes of degree one at the critical point and it is the solution of the following set of coupled differential equations

$$\frac{d\delta^{(ij)}}{dy}(y) \quad = \quad -\frac{e(y)}{y}\left[\hat{f}^{(ij)}(y)+\sum_{k=0}^{d}\left(\delta^{(ik)}(y)\frac{\partial \hat{f}^{(j)}}{\partial z^{(k)}}(y)+\frac{\partial \hat{f}^{(i)}}{\partial z^{(k)}}(y)\delta^{(kj)}(y)\right)\right].$$

The expressions for $e(y)$, $\hat{f}^{(ij)}(y))$ and $\hat{f}^{(i)}(y)$ as well as the initial conditions are given in Section 2.4.2.

As explained in more detail in the sequel, the dimension of the differential system can be quite large. Fortunately, it is possible to express α as a function of the degree distributions and the parameters of the critical point. We discuss this alternative computation in Section 2.4.3.

Lemma 3. [Alternative Expression for α] Consider transmission over a BEC using random elements from an ensemble LDPC$(n, \lambda(x), \rho(x))$. Then the scaling parameter in Lemma 1 is given by

$$\alpha = \left(\frac{\rho(\bar{x}^*)^2-\rho(\bar{x}^{*2})+\rho'(\bar{x}^*)(1-2x^*\rho(\bar{x}^*))-\bar{x}^{*2}\rho'(\bar{x}^{*2})}{\Lambda'(1)\lambda(y^*)^2\rho'(\bar{x}^*)^2}+\right.$$
$$\left.\frac{\epsilon^{*2}\lambda(y^*)^2-\epsilon^{*2}\lambda(y^{*2})-y^{*2}\epsilon^{*2}\lambda'(y^{*2})}{\Lambda'(1)\lambda(y^*)^2}\right)^{1/2},$$

where ϵ^* is the threshold, y^* the non-zero root of $r_1(y)=0$, $x^*=\epsilon^*\lambda(y^*)$ and finally $\bar{x}^*=1-x^*$. For the case of regular ensembles $\lambda(x)=x^{\mathtt{l}-1}$ and $\rho(x)=x^{\mathtt{r}-1}$ the expression simplifies to

$$\alpha=\epsilon^*\sqrt{\frac{\mathtt{l}-1}{\mathtt{l}}\Big(\frac{1}{x^*}-\frac{1}{y^*}\Big)}.$$

The Scaling Parameter β

The refinement of the scaling law in Conjecture 1 is explained in Section 2.5. The parameter β describes the speed at which the finite-length threshold (the point at which the block error probability equals one-half) converges to the asymptotic threshold.

Conjecture 2. [Scaling Parameter β] Consider transmission over a BEC using random elements from an ensemble LDPC$(n, \lambda(x), \rho(x))$. Then the scaling parameter in Conjecture 1 is given by

$$\beta = \quad \left(\frac{\epsilon^{*4}r_2^{*2}(\epsilon^*\lambda'(y^*)^2r_2^*-x^*(\lambda''(y^*)r_2^*+\lambda'(y^*)x^*))^2}{\Lambda'(1)^2\rho'(\bar{x}^*)^3x^{*10}(2\epsilon^*\lambda'(y^*)^2r_3^*-\lambda''(y^*)r_2^*x^*)}\right)^{1/3}, \tag{2.7}$$

where for $i \geq 2$

$$r_i^* = \sum_{m\geq j\geq i}(-1)^{i+j}\binom{j-1}{i-1}\binom{m-1}{j-1}\rho_m(\epsilon^*\lambda(y^*))^j.$$

For regular ensembles $\lambda(x) = x^{\mathtt{l}-1}$ and $\rho(x) = x^{\mathtt{r}-1}$ this expression simplify to

$$\beta \quad = \epsilon^* \left(\tfrac{\mathtt{l}-2}{\mathtt{l}x^*y^*}\right)^{2/3} \left(\tfrac{1}{(\mathtt{l}-1)} + \tfrac{(\mathtt{r}-2)x^*}{1-x^*} - 2\right)^{-1/3}, \tag{2.8}$$

where ϵ^* is the threshold, y^* the non-zero root of $r_1(y) = 0$, $x^* = \epsilon^*\lambda(y^*)$ and finally $\bar{x}^* = 1 - x^*$.

An applet that computes from the degree distributions β as given in Lemma 2 and α from both covariance evolution as in Lemma 2 and also the alternative expression given in Lemma 3 is online at the following address `http://lthiserv5.epfl.ch/scalingbec/`

2.3.3 Informal Justification

Consider the ensemble LDPC$(n, \lambda(x), \rho(x))$, transmission over a BEC of erasure probability ϵ, and decoding using the PA (Section 2.2). We have already explained that one can follow the progress of the decoding by observing the number of degree one check nodes in the residual graph R_1. If $\mathrm{R}_1 = 0$ before the residual graph is empty then the decoder fails. In Fig. 2.2 we represent the number of degree-one check nodes in the decoder as a function of the number of unknown variables in the residual graph for the ensemble LDPC$(n, \lambda(x) = x^2, \rho(x) = x^5)$ for $n = 2048$ and $n = 8192$. Each trajectory corresponds to a particular choice of the graph and

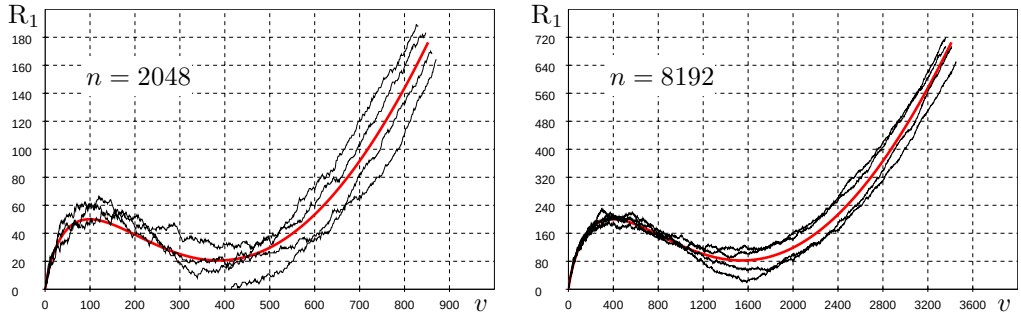


Figure 2.2: Decoding trajectories for the ensemble LDPC$(n, \lambda(x) = x^2, \rho(x) = x^5)$ for $n = 2048$ and $n = 8192$. The transmission is over a BEC of erasure probability $\epsilon = 0.415$.

the channel realization. We have seen in Section 2.2 that these trajectories closely follow the asymptotic expected value (thick curve given by the density evolution equations) and we will

show that their standard deviation is of order $\sqrt{n}$. Consider now the decoding process at the channel parameter ϵ close to ϵ^*. If $\epsilon = \epsilon^*$ then at the critical point (the point at which the density evolution curve hits the zero axis) the expected number of degree-one check nodes is zero. Assume now that we vary ϵ slightly. From the density evolution equation (2.5) we see that the expected change in the number of degree-one check nodes R_1 at the critical point is

$$\left.\frac{\partial \mathrm{R}_1}{\partial \epsilon}\right|_{y=y^*;\epsilon=\epsilon^*} = -n\Lambda'(1)\epsilon^*\lambda(y^*)^2\rho'(1-\epsilon^*\lambda(y^*)). \tag{2.9}$$

If we vary ϵ so that $\Delta\epsilon$ is of order $\Theta(1)$, then we conclude from (2.9) that the expected number of degree-one check nodes at the critical point is of order $\Theta(n)$. Since the standard deviation is of order $\Theta(\sqrt{n})$, then with high probability the decoding process will either succeed (if $(\epsilon-\epsilon^*) < 0$) or fail (if $(\epsilon - \epsilon^*) > 0$). The interesting scaling happens if we choose our variation of ϵ in such a way that $\Delta\epsilon = z/\sqrt{n}$, where z is a constant. In this case the expected gap at the critical point scales in the same way as the standard deviation and one would expect that the probability of error stays constant. Varying now the constant z will give rise to the scaling function $f(z)$, cf. equation (2.1).

We will further see that the distribution of the degrees in the residual graph (the state) at any time before hitting the $\mathrm{R}_1 = 0$ plane is asymptotically Gaussian and that the evolution of its covariance matrix is governed by a set of differential equations in the same way as the mean is in density evolution. We will therefore call these equations the *covariance evolution* equations.

Once one knows the variance of the number of degree-one check nodes throughout the decoding, one can quantify the probability for the process to hit the $\mathrm{R}_1 = 0$ plane as follows. Stop density and covariance evolution when the fraction of variable nodes reaches the critical value ν^* (the fractional size of the residual graph at the threshold). At this point the probability distribution of the state is well approximated by a Gaussian with a given mean and covariance for $\mathrm{R}_1 \geq 0$ (while it is obviously 0 for $\mathrm{R}_1 < 0$). Estimate the survival probability (i.e. the probability of not hitting the $\mathrm{R}_1 = 0$ plane at any time) by summing the Gaussian distribution over $\mathrm{R}_1 \geq 0$. Obviously this integral can be expressed in terms of a Q-function.

We will see that the above description indeed leads to the scaling behavior as stated in Lemma 1. Where does the shift in Conjecture 1 come from? It is easy to understand that we were a bit optimistic (i.e., we underestimated the error probability) in the above calculation: We correctly excluded from the sum the part of the Gaussian distribution lying in the $\mathrm{R}_1 < 0$ half-space – trajectories contributing to this part must have hit the $\mathrm{R}_1 = 0$ plane at some point in the past. On the other hand, we cannot be certain that trajectories such that $\mathrm{R}_1 > 0$ at the critical point

didn't hit the $\mathrm{R}_1 = 0$ plane at some time in the past and bounced back (or will not hit it at some later point). We refer to Section 2.5 for an in-depth discussion on how to estimate this effect.

2.4 Analysis

2.4.1 Finite Dimensional Markov Process

Consider a family of Markov chains $X_{n,0}, X_{n,1}, \ldots, X_{n,t}, \ldots$ parametrized by $n \in \mathbb{N}$ and taking values in $\mathbb{Z}^{d+1}$. For iterative decoding applications, n will represent the blocklength. We drop the subscript n hereafter. Let the transition probability be

$$\mathrm{P}(X_{t+1} = x' | X_t = x) = W(x' - x|x)\,, \tag{2.10}$$

and the initial condition $X_0 \in \mathbb{Z}^{d+1}$. We will denote the $d+1$ coordinates of the state x as

$$(x^{(0)}, x^{(1)}, \ldots, x^{(d)}) = x \in \mathbb{Z}^{d+1}\,. \tag{2.11}$$

We denote the corresponding random variable by $(X^{(0)}, X^{(1)}, \ldots, X^{(d)})$.

In the following we shall always be interested in times $t < \kappa_0\, n$ for a positive constant κ_0 (we reserve the symbols $\kappa_1, \kappa_2, \ldots$ for numerical constants which we assume do not to depend upon n). We shall moreover assume the following regularity properties of the Markov chain:

1. The chain makes finite jumps. In other words, there exists a $\kappa_1 > 0$ such that $|X^{(i)}_{t+1} - X^{(i)}_t| < \kappa_1$ almost surely.

2. The transition probabilities have a smooth $n \to \infty$ limit. In practice there exist functions $\widehat{W} : \mathbb{Z}^{d+1} \times \mathbb{R}^{d+1} \to \mathbb{R}_+$ and a positive constant κ_2 such that

$$|W(\Delta|x) - \widehat{W}(\Delta|x/n)| < \kappa_2/n\,. \tag{2.12}$$

   Clearly, we have $\sum_\Delta \widehat{W}(\Delta|x/n) = 1$. We shall moreover assume $\widehat{W}(\Delta|z)$ to be $C^2(\mathbb{R}^{d+1})$ with respect to its second argument and to have bounded first and second derivatives.

3. The process has a finite range on the n scale. In practice, there exists $\kappa_3 > 0$ such that $|X^{(i)}_t| < \kappa_3\, n$ almost surely.

Under these hypothesis the distribution of X_t is well described by a Gaussian whose mean and variance can be obtained by solving some ordinary differential equations. In order to state

this fact in a more precise fashion, we need some additional notation. We denote by $\overline{X}_t \equiv \mathbb{E}[X_t]$ the average of X_t and $D_t^{(ij)} \equiv \mathbb{E}[X_t^{(i)}; X_t^{(j)}] \equiv \mathbb{E}[X_t^{(i)} X_t^{(j)}] - \mathbb{E}[X_t^{(i)}]\mathbb{E}[X_t^{(j)}]$ its covariance. We need furthermore the first two moments of the transition rates $W(\Delta|x)$:

$$f^{(i)}(x) \equiv \sum_{\Delta} \Delta_i \, W(\Delta|x)\,, \tag{2.13}$$

$$f^{(ij)}(x) \equiv \sum_{\Delta} \Delta_i \Delta_j \, W(\Delta|x) - f^{(i)}(x) f^{(j)}(x)\,, \tag{2.14}$$

with $i, j \in \{0, \ldots, d\}$. We shall call $\hat{f}^{(i)}(z)$, $\hat{f}^{(ij)}(z)$ the analogous quantities for the limiting rates $\widehat{W}(\Delta|z)$.

Finally, let $\overline{z}(\tau) \in \mathbb{R}^{d+1}$ and $\delta^{(ij)}(\tau) \in \mathbb{R}$, for $\tau \in \mathbb{R}_+$ and $i, j \in \{0, \ldots, d\}$, denote the solution of

$$\frac{d\overline{z}^{(i)}}{d\tau}(\tau) = \hat{f}^{(i)}(\overline{z}(\tau))\,, \tag{2.15}$$

$$\frac{d\delta^{(ij)}}{d\tau}(\tau) = \hat{f}^{(ij)}(\overline{z}(\tau)) + \sum_{k=0}^{d} \left[\delta^{(ik)}(\tau) \left.\frac{\partial \hat{f}^{(j)}}{\partial z^{(k)}}\right|_{\overline{z}(\tau)} + \left.\frac{\partial \hat{f}^{(i)}}{\partial z^{(k)}}\right|_{\overline{z}(\tau)} \delta^{(kj)}(\tau) \right] \tag{2.16}$$

with initial conditions $\overline{z}(0) = \mathbb{E}[X_0]/n$ and $\delta^{(ij)}(0) = D_0^{(ij)}/n$.

Lemma 4. *Under the conditions stated above the following results hold (here we use the symbols $\Omega_0, \Omega_1, \ldots$, for constants (independent of n) which we prove to exist):*

I. X_t concentrates on the n scale. In formula, there exist $\Omega_0 > 0$, such that

$$\mathrm{P}\{|X_t^{(i)} - \overline{X}_t^{(i)}| \geq \rho\} \leq 2\, e^{-\frac{\rho^2}{2\Omega_0 t}}. \tag{2.17}$$

II. The average and covariance of X_t are accurately tracked by $\overline{z}(\tau)$ and $\delta^{(ij)}(\tau)$. More precisely, there exist constants $\Omega_1, \Omega_2 > 0$, such that

$$\left|\frac{1}{n}\overline{X}_t^{(i)} - \overline{z}^{(i)}(t/n)\right| \leq \frac{\Omega_1}{n}\,, \tag{2.18}$$

$$\left|\frac{1}{n} D_t^{(ij)} - \delta^{(ij)}(t/n)\right| \leq \frac{\Omega_2}{\sqrt{n}}. \tag{2.19}$$

III. The variable $(X_t - \overline{X}_t)/\sqrt{n}$ converges weakly to a $(d+1)$-dimensional Gaussian with variance $\delta^{(ij)}(t/n)$. More precisely, define the logarithmic moment generating function

$$\Lambda_t(\lambda) \equiv \log \mathbb{E} \, \exp\left[\frac{1}{\sqrt{n}} \, \lambda \cdot (X_t - \overline{X}_t)\right], \tag{2.20}$$

for $\lambda \in \mathbb{R}^{d+1}$. Then there exist a function $\lambda \mapsto \Omega_4(\lambda) \in \mathbb{R}_+$, such that

$$\left| \Lambda_t(\lambda) - \frac{1}{2} \sum_{ij} \delta^{(ij)}(t/n) \lambda_i \lambda_j \right| \leq \frac{\Omega_4(\lambda)}{\sqrt{n}} \,. \tag{2.21}$$

The situation investigated here can be regarded as a discrete analogous of the Friedlin-Wentzell theory of random perturbations of dynamical systems [24]. The proof is given in Appendix B.

2.4.2 Covariance Evolution

Consider now the ensemble LDPC$(n, \lambda(x), \rho(x))$, transmission over the BEC of erasure probability ϵ and decoding using the Peeling algorithm. We define the state of the decoder as in Section 2.2 $X_t = (\mathrm{R}_1, \cdots, \mathrm{R}_{\mathtt{r}_{\max}-1}, \mathrm{L}_1, \cdots, \mathrm{L}_{\mathtt{l}_{\max}})$. It describes the residual graph of the decoder after t decoding rounds.

We are interested in solving the system of equations (2.16) for our particular case

$$\frac{d\delta^{(ij)}}{d\tau}(\tau) \quad = \quad \hat{f}^{(ij)}(\overline{z}(\tau)) + \sum_{k=0}^{d} \left[\delta^{(ik)}(\tau) \left. \frac{\partial \hat{f}^{(j)}}{\partial z^{(k)}} \right|_{\overline{z}(\tau)} + \left. \frac{\partial \hat{f}^{(i)}}{\partial z^{(k)}} \right|_{\overline{z}(\tau)} \delta^{(kj)}(\tau) \right] ,$$

where $\tau = t/n$ represents the normalized time and $d = \mathtt{l}_{\max} + \mathtt{r}_{\max} - 1$. In [7], equations (2.15) were already solved and the solution vector $\overline{z}(\tau)$ was parametrized with the variable y such that $dy/d\tau = -y/e(y)$ and $e(y) = \epsilon y \lambda(y)$. Let us make the same change of variable, equations (2.16) become

$$\frac{d\delta^{(ij)}}{dy}(y) \quad = \quad -\frac{e(y)}{y} \left[\hat{f}^{(ij)}(y) + \sum_{k=0}^{d} \left(\delta^{(ik)}(y) \frac{\partial \hat{f}^{(j)}}{\partial z^{(k)}}(y) + \frac{\partial \hat{f}^{(i)}}{\partial z^{(k)}}(y) \delta^{(kj)}(y) \right) \right] .$$

y also has a practical meaning as it represents the fraction of erased messages sent from the check nodes to the variable nodes in the equivalent BP decoder.

Consider the normalized quantities to describe the graph $l_i = \mathrm{L}_i/\xi$ with $1 \leq i \leq \mathtt{l}_{\max}$ and $r_i = \mathrm{R}_i/\xi$ with $1 \leq i \leq \mathtt{r}_{\max}$. ξ the initial number of edges in the graph is equal to $n\Lambda'(1)$. During the decoding a fraction $e = \sum_{i=1}^{\mathtt{l}_{\max}} l_i = \sum_{i=1}^{\mathtt{r}_{\max}} r_i$ of edges remains in the residual graph. An edge taken at random in the residual graph, has probability $p_i = \frac{l_i}{e}$ of being connected to a variable node of degree i. Similarly, it has probability $q_i = \frac{r_i}{e}$ of being connected to a check node of degree i. Define further $a = \sum_{i=1}^{\mathtt{l}_{\max}} i p_i$.

In one decoding step, a variable node connected to a degree one check node is pealed off from the graph. This results in $(i-1)$ other edges being removed from the graph, where i is the degree

of the chosen variable node. In the large blocklength limit, the joint probability that the i edges are connected to u_j degree j check nodes, $j \in \{1, \cdots, \mathtt{r}_{\max}\}$, tends to a product distribution and can be computed easily,

$$w_i(u_1, u_2, \cdots, u_{\mathtt{r}_{\max}}) = \binom{i-1}{u_1 - 1, u_2, \cdots, u_{\mathtt{r}_{\max}}} q_1^{u_1 - 1} q_2^{u_2} \cdots q_{\mathtt{r}_{\max}}^{u_{\mathtt{r}_{\max}}}.$$

This probability is conditioned on the fact that the removed variable node is of degree i, which happens with probability p_i. We can write down the generating function

$$\begin{aligned} W_i(x_1, x_2, \cdots, x_{\mathtt{r}_{\max}}) &= \sum_{u_1, \cdots, u_{\mathtt{r}_{\max}}} w_i(u_1, \cdots, u_{\mathtt{r}_{\max}}) x_1^{u_1} \cdots x_{\mathtt{r}_{\max}}^{u_{\mathtt{r}_{\max}}} \\ &= x_1 (x_1 q_1 + x_2 q_2 + \cdots + x_{\mathtt{r}_{\max}} q_{\mathtt{r}_{\max}})^{i-1}. \end{aligned}$$

We can now compute the different terms needed to solve the differential system. The indices in superscript will be replaced to be slightly more informative. For example, the drift $\hat{f}^{(0)}$ will be replaced by $\hat{f}^{l_1}$.

Local Drifts

$\hat{f}^{(r_j)}$ represents the expected change in the number of edges connected to check nodes of degree j in the residual graph when we perform one step of the decoding. For $j \in \{2, .., \mathtt{r}_{\max}\}$, we obtain

$$\begin{aligned} \hat{f}^{(r_j)} &= j \sum_{i=2}^{\mathtt{l}_{\max}} p_i \sum_{u_1, .., u_{\mathtt{r}_{\max}}} w_i(u_1, .., u_{\mathtt{r}})(u_{j+1} - u_j) \\ &= j \sum_{i=2}^{\mathtt{l}_{\max}} p_i (i-1)(q_{j+1} - q_j) \\ &= j(q_{j+1} - q_j) \left(\sum_{i=2}^{\mathtt{l}_{\max}} i p_i - 1 \right) \\ &= j(r_{j+1} - r_j) \frac{(a-1)}{e}, \end{aligned}$$

while for degree one check nodes, we obtain

$$\begin{aligned} \hat{f}^{(r_1)} &= \sum_{i=2}^{\mathtt{l}_{\max}} p_i \sum_{u_1, .., u_{\mathtt{r}_{\max}}} w_i(u_1, .., u_{\mathtt{r}_{\max}})(u_2 - u_1) \\ &= -1 + (r_2 - r_1) \frac{(a-1)}{e}. \end{aligned}$$

For the variable nodes, we can write for $i \in 1, \cdots, \mathtt{l}_{\max}$,

$$\hat{f}^{(l_i)} = -i p_i = -\frac{i l_i}{e}.$$

These terms were used in [7] to find solution of density evolution that we stated in section 2.2.

$$\begin{aligned} l_i(y) &= \epsilon\lambda_i y^i \\ r_1(y) &= \epsilon\lambda(y)[y-1+\rho(1-\epsilon\lambda(y))] \\ r_i(y) &= \sum_{m\geq j\geq i}(-1)^{i+j}\binom{j-1}{i-1}\binom{m-1}{j-1}\rho_m(\epsilon\lambda(y))^j \end{aligned}$$

with

$$\begin{aligned} e(y) = &\quad \textstyle\sum_{i=1}^{\mathtt{l}_{\max}} l_i(y) &&= \epsilon y\lambda(y) \\ a(y) = &\quad \frac{\sum_{i=1}^{\mathtt{l}_{\max}} i l_i(y)}{e(y)} &&= 1+\frac{y\lambda'(y)}{\lambda(y)}, \end{aligned}$$

where $y \in (0,1)$ corresponds to the probability (fraction) that an edge carries an erased message to a variable node in the equivalent BP decoder.

Local Covariances

In a way similar to the calculation of the drifts, the local covariance terms can also be computed. For check nodes of degree $j \in \{1,\cdots,\mathtt{r}_{\max}\}$,

$$\begin{aligned} \hat{f}^{(r_j r_j)} &= j^2 \sum_{i=2}^{\mathtt{l}_{\max}} p_i \sum_{u_1,..,u_{\mathtt{r}_{\max}}} w_i(u_1,..,u_{\mathtt{r}_{\max}})(u_{j+1}-u_j)^2 - \left(\hat{f}^{(r_j)}\right)^2 \\ &= j^2\left(\left(\frac{y^2\lambda''(y)}{\lambda(y)} - \frac{y^2\lambda'(y)^2}{\lambda(y)^2}\right)\frac{(r_{j+1}-r_j)^2}{e(y)^2} + \frac{y\lambda'(y)}{\lambda(y)}\frac{(r_{j+1}+r_j)}{e(y)}\right). \end{aligned}$$

We will omit the other calculations and just write the general formula, for i and $j \in \{1,\cdots,\mathtt{r}_{\max}\}$

$$\begin{aligned} \hat{f}^{(r_i r_j)} &= ij\left(\frac{y^2\lambda''(y)}{\lambda(y)} - \frac{y^2\lambda'(y)^2}{\lambda(y)^2}\right)\frac{(r_{i+1}-r_i)(r_{j+1}-r_j)}{e(y)^2} \\ &\quad + ij\frac{y\lambda'(y)}{\lambda(y)}\left(\mathbb{1}_{\{i=j\}}\frac{(r_{i+1}+r_i)}{e(y)} - \mathbb{1}_{\{j=i+1\}}\frac{r_j}{e(y)} - \mathbb{1}_{\{i=j+1\}}\frac{r_i}{e(y)}\right), \end{aligned}$$

where $\mathbb{1}_{\{i=j\}}$ is the indicator function, equal to 1 if the condition $(i=j)$ is fulfilled and 0 otherwise. For variable nodes, we obtain for $i,j \in 1,..,\mathtt{l}_{\max}$,

$$\hat{f}^{(l_i l_j)} = ij\frac{l_i}{e(y)}\left(\mathbb{1}_{\{i=j\}} - \frac{l_j}{e(y)}\right).$$

Now for the cross terms between variable and check degrees, we have, for $i \in 1,..,\mathtt{l}_{\max}$ and $j \in 1,..,\mathtt{r}_{\max}$

$$\hat{f}^{(l_i r_j)} = \left(\frac{y\lambda'(y)}{\lambda(y)} - (i-1)\right) ij\frac{l_i(r_{j+1}-r_j)}{e(y)^2}.$$

Initial Covariances

In order to solve the differential system, we still have to specify the initial conditions. These are the covariances of the number of edges of each degree at the start of the decoding divided by the total number of edges in the original graph. For the variables, we have for $i \in \{1, ..\mathtt{l}_{\max}\}$

$$\delta^{l_i l_j}(y=1) \quad = \quad \mathbb{1}_{\{i=j\}}(i\lambda_i \epsilon\bar{\epsilon}).$$

For the check nodes, we obtain, for $j \in \{1, ..\mathtt{r}_{\max}\}$

$$\delta^{r_j r_j}(y=1) \quad = \quad \sum_{k=1}^{\mathtt{r}_{\max}} \frac{\rho_k}{k}\left(jA_j^k - (A_j^k)^2\right) + \lambda'(1)\sum_{i,k}^{\mathtt{r}_{\max}} \rho_i\rho_k\left(B_{j,j}^{i,k} - A_j^i A_j^k\right)$$

where $A_j^k = k\binom{k-1}{j-1}\epsilon^j(1-\epsilon)^{k-j}$ and

$$\begin{aligned} B_{j,l}^{i,k} \quad &= \quad jl\left(\epsilon\left(\binom{i-1}{j-1}\epsilon^{j-1}\bar{\epsilon}^{i-j}\binom{k-1}{l-1}\epsilon^{l-1}\bar{\epsilon}^{k-l}\right)\right. \\ &\quad \left. + \bar{\epsilon}\left(\binom{i-1}{j}\epsilon^{j}\bar{\epsilon}^{i-j-1}\binom{k-1}{l}\epsilon^{l}\bar{\epsilon}^{k-l-1}\right)\right). \end{aligned}$$

For cross terms, we get

$$\delta^{r_j r_l}(y=1) \quad = \quad \lambda'(1)\sum_{i,k}^{\mathtt{r}_{\max}} \rho_i\rho_k\left(B_{j,l}^{i,k} - A_j^i A_l^k\right) - \sum_{i}^{\mathtt{r}_{\max}} \frac{\rho_i}{i}\left(A_j^i A_l^i\right).$$

For terms involving variable and check degrees, we get

$$\delta^{r_j l_i}(y=1) \quad = \quad \sum_{l} \rho_l\lambda_i\left(ji\epsilon\binom{l-1}{j-1}\epsilon^{j-1}\bar{\epsilon}^{l-j} - i\epsilon A_j^l\right).$$

Derivatives of Local Drifts

The derivatives of the drifts are the last terms still needed for this computation. In our definition of the states, we have described the constraint $\sum_{i=1}^{\mathtt{l}_{\max}} l_i = \sum_{j=1}^{\mathtt{r}_{\max}} r_j$ that the fractions had to obey. Up to this point of the paper, this constraint didn't affect our expressions, but it will do so for the derivatives of the drifts.

Consider $i \in \{1, \cdots \mathtt{r}_{\max} - 2\}$, $j \in \{1, \cdots, \mathtt{r}_{\max} - 1\}$ and $k \in \{1, \cdots, \mathtt{l}_{\max}\}$ then the

derivatives of the drifts are taken in the usual way as for an unconstrained space, giving,

$$\begin{aligned}
\frac{\partial \hat{f}^{(r_i)}}{\partial r_i} &= -i\frac{(a-1)}{e(y)} \\
\frac{\partial \hat{f}^{(r_i)}}{\partial r_{i+1}} &= i\frac{(a-1)}{e(y)} \\
\frac{\partial \hat{f}^{(r_i)}}{\partial r_j} &= 0, \quad \text{for} \quad j \neq i \quad \text{and} \quad j \neq i+1 \\
\frac{\partial \hat{f}^{(r_i)}}{\partial l_k} &= -i(r_{i+1}-r_i)\frac{(2a-k-1)}{e^2}.
\end{aligned}$$

However, the constraint being fulfilled by setting the fraction $r_{\mathtt{r}_{\max}} = \sum_{i=1}^{\mathtt{l}_{\max}} l_i - \sum_{j=1}^{\mathtt{r}_{\max}-1} r_j$, we have that for the last check fraction in our states $\mathtt{r}_{\max}-1$, the derivatives for $j \in \{1,..,\mathtt{r}_{\max}-2\}$ and $k \in \{1,\cdots,\mathtt{l}_{\max}\}$ are

$$\begin{aligned}
\frac{\partial \hat{f}^{(r_{\mathtt{r}_{\max}-1})}}{\partial r_j} &= -(\mathtt{r}_{\max}-1)\frac{(a-1)}{e} \\
\frac{\partial \hat{f}^{(r_{\mathtt{r}_{\max}-1})}}{\partial r_{\mathtt{r}_{\max}-1}} &= -2(\mathtt{r}_{\max}-1)\frac{(a-1)}{e} \\
\frac{\partial \hat{f}^{(\mathtt{r}_{\max}-1)}}{\partial l_k} &= (\mathtt{r}_{\max}-1)\left(\frac{k-a}{e} + (2a-k-1)\frac{(\sum_{i=1}^{\mathtt{r}_{\max}-1} r_i + r_{\mathtt{r}_{\max}-1})}{e^2}\right).
\end{aligned}$$

For the drifts of the variable node degrees, we have the following derivatives, for $i,j \in \{1,..,\mathtt{l}_{\max}\}$, $i \neq j$, and $k \in \{1,..,\mathtt{r}_{\max}-1\}$

$$\begin{aligned}
\frac{\partial \hat{f}^{(l_i)}}{\partial l_i} &= \frac{il_i}{e(y)^2} - \frac{i}{e(y)} \\
\frac{\partial \hat{f}^{(l_i)}}{\partial l_j} &= \frac{il_i}{e(y)^2} \\
\frac{\partial \hat{f}^{(l_i)}}{\partial r_k} &= 0.
\end{aligned}$$

We have now computed all the terms needed. We can solve the system and obtain the covariance matrix of the state evolving through the decoding. At the threshold ϵ^*, we have $r_1(y^*) = 0$, the asymptotic mean of the number of degree one check nodes is 0. The normalized asymptotic variance of R_1, the number of degree one check nodes is equal to $\delta^{r_1,r_1}(y^*)$. For long lengths, the variance of the number of degree one check nodes is therefore well approximated by $n\Lambda'(1)\delta^{r_1,r_1}(y^*)$. We have this particular factor in front as R_1 was normalized by $\xi = n\Lambda'(1)$ to obtain r_1.

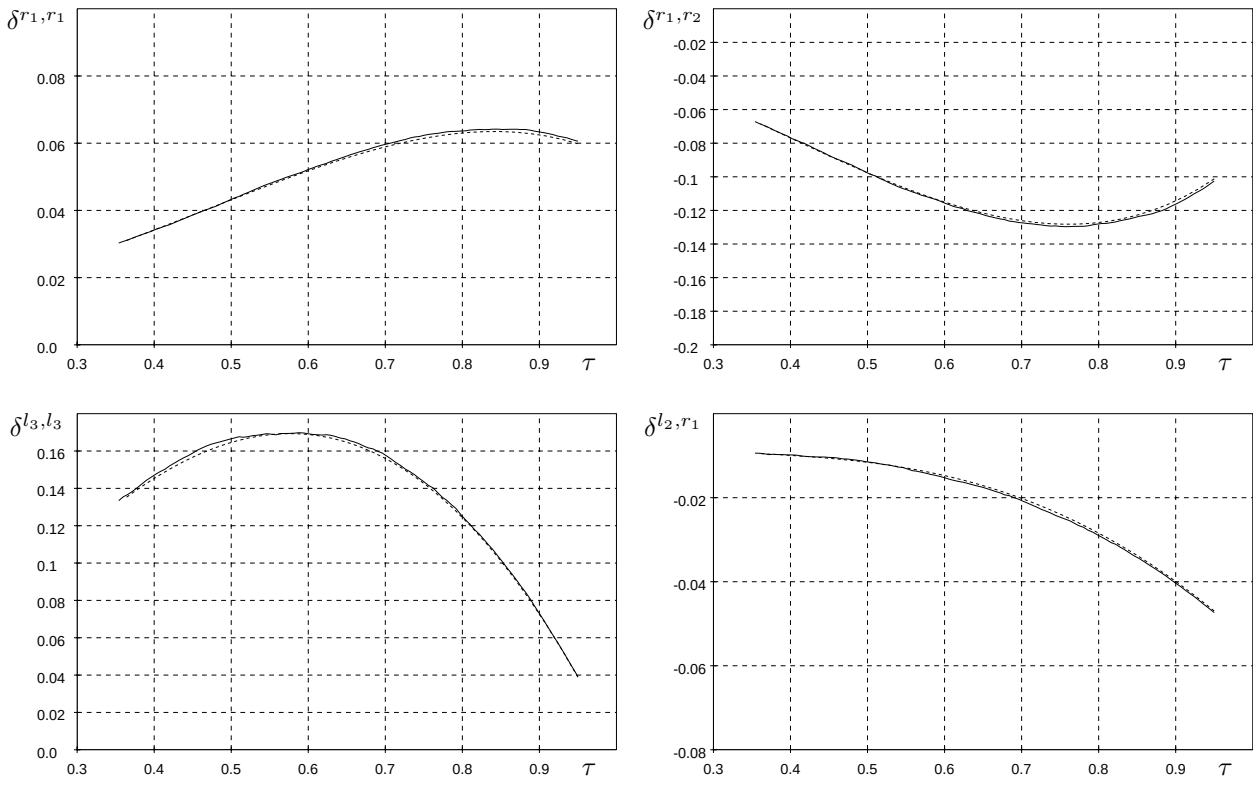


Figure 2.3: Covariance evolution through the decoding as a function of the normalized time. The code ensemble we are using has the degree distributions $\Lambda(x) = 0.8x^2 + 0.2x^3$ and $P(x) = 0.8x^2 + 0.2x^3$. The threshold is $\epsilon^* = 0.989787$ and we are performing this experiment for $\epsilon = 0.95$. The dashed curve represents, the solution of covariance evolution while the solid curve represents the empirical covariances obtained by simulating codes of length 4000.

If we are operating at an erasure probability $\epsilon = \epsilon^* - z/\sqrt{n}$ with z a constant and n the increasing blocklength, then the mean of the number of degree one check nodes is close to

$$(\epsilon - \epsilon^*) \left.\frac{\partial \mathrm{R}_1}{\partial \epsilon}\right|_{y^*;\epsilon^*} = n\Lambda'(1)(\epsilon^* - \epsilon)\epsilon^*\lambda(y^*)^2\rho'(1-\epsilon^*\lambda(y^*)). \tag{2.22}$$

The distribution of R_1 tends to a Gaussian. We therefore, obtain our scaling formula

$$\begin{aligned}
\mathrm{P}_{\mathrm{B},\gamma}(n,\lambda,\rho,\epsilon) &= Q\left(\frac{n\Lambda'(1)(\epsilon^*-\epsilon)\epsilon^*\lambda(y^*)^2\rho'(1-\epsilon^*\lambda(y^*))}{\sqrt{n\Lambda'(1)\delta^{r_1,r_1}(y^*)}}\right)(1+o_n(1)),\\
&= Q\left(\frac{\sqrt{n}(\epsilon^*-\epsilon)}{\frac{\sqrt{\delta^{r_1,r_1}(y^*)}}{\sqrt{\Lambda'(1)}\epsilon^*\lambda(y^*)^2\rho'(1-\epsilon^*\lambda(y^*))}}\right)(1+o_n(1)),\\
&= Q\left(\frac{\sqrt{n}(\epsilon^*-\epsilon)}{\alpha}\right)(1+o_n(1)),\\
&= Q\left(\frac{z}{\alpha}\right)(1+o_n(1)),
\end{aligned} \tag{2.23}$$

with α as stated in Lemma 2

$$\alpha = \frac{\sqrt{\delta^{r_1,r_1}(y^*)}}{\sqrt{\Lambda'(1)}\epsilon^*\lambda(y^*)^2\rho'(1-\epsilon^*\lambda(y^*))}.$$

If one considers the bit error probability asymptotically for channel erasure probabilities close to the threshold, then the typical size of the failure is $n\nu^*$, with $\nu^* = \epsilon^*\Lambda(y^*)$. We can write

$$\mathrm{P}_{\mathrm{b},\gamma}(n,\lambda,\rho,\epsilon) = \nu^* Q\left(\frac{z}{\alpha}\right)(1+o_n(1)).$$

It is worth noting at this point, that we didn't show yet that the decoding process we are studying fulfills the conditions of Lemma 4. Conditions 1. and 3. are obviously verified. Remains condition 2. It is not directly verified, but we show in Appendix C. that the decoding process can be extended to a process that fulfills condition 2. and that in this case the probability of error is indeed dominated by the expression given in Lemma 1 which completes the proof.

2.4.3 Alternative Computation for the Scaling Parameter

Consider the differential system of covariance evolution (equations (2.16)). In order to obtain the scaling parameter α, one needs to solve this system which is of high dimension. The number of equations being equal to the square of the number of variable node degrees plus the largest check

node degree minus one. As en example, for an ensemble with 5 different variable node degrees and $\mathtt{r}_{\max} = 30$, the number of coupled equations in covariance evolution is $(5+29)^2 = 1156$. The computation of the scaling parameter can therefore become a challenging task. This is of particular concern if one is interested in using this approach as the basis of a finite length optimization in which the scaling parameter for a large number of codes has to be computed quickly. The main result in this section is to show that it is possible to compute the scaling parameters without explicitly solving covariance evolution.

In order to show result the result of Lemma 3, we will first compute the variance of the messages exchanged in the BP decoder and later show that this quantity can be related to α.

Variance of the Messages

Consider the ensemble LDPC$(n, \lambda(x), \rho(x))$ and assume that transmission takes place over a BEC of parameter ϵ. Perform ℓ iterations of BP and then set $\mu_i^{(\ell)}$ equal to 1 if the message sent out along edge i from variable to check node is an erasure and 0, otherwise. Consider the variance of these messages in the limit of large blocklengths. More precisely, consider

$$\mathcal{V}^{(\ell)} = \lim_{n\to\infty} \frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2] - \mathbb{E}[(\sum_i \mu_i^{(\ell)})]^2}{n\Lambda'(1)},$$

where the expectations are taken with respect to both the graph and the channel realizations. Lemma 9 in Appendix D contains an analytic expression for this quantity as a function of the degree distributions $\lambda(x)$ and $\rho(x)$, the channel parameter ϵ and the number of iterations ℓ. Let us consider this variance as a function of the parameter ϵ and the number of iterations ℓ. Fig. 2.4 shows the result of this evaluation for the case $(\Lambda(x) = \frac{2}{5}x^2 + \frac{3}{5}x^3; P(x) = \frac{3}{10}x^2 + \frac{7}{10}x^3)$. The threshold (critical value) for this example is $\epsilon^* \approx 0.8495897455$. This value is indicated as a vertical line in the figure. As we can see from this figure, the variance is a unimodal function of the channel parameter. It is zero for the extremal values of ϵ (either all messages are known or all are erased) and it takes on a maximum value for a parameter of ϵ which approaches the critical value ϵ^* as ℓ increases. Further, for increasing ℓ the maximum value of the variance increases. The limit of these curves for ℓ tending to infinity is also shown (bold curve). It has the following behavior: the variance is zero below the threshold; above the threshold it is positive and in particular it grows beyond any bound as the curve approaches the threshold value. In Appendix D we state the exact form of the limiting curve. We show that for ϵ larger than the

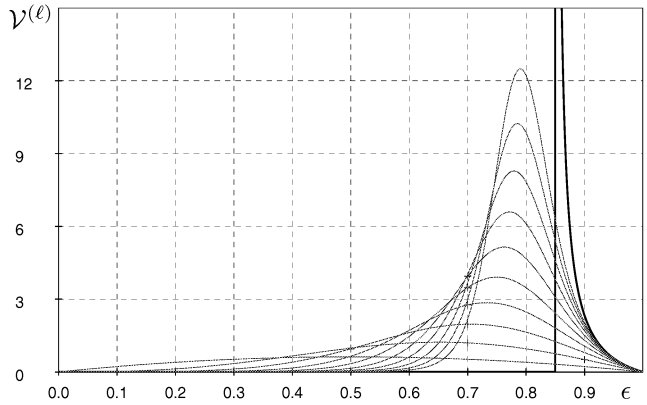


Figure 2.4: The variance as a function of ϵ and $\ell = 0, \cdots, 9$ for $(\Lambda(x) = \frac{2}{5}x^2 + \frac{3}{5}x^3; P(x) = \frac{3}{10}x^2 + \frac{7}{10}x^3)$. Also shown is the limiting curve $\mathcal{V} = \lim_{\ell\to\infty} \mathcal{V}^{(\ell)}$.

threshold ϵ^* it has the form

$$\mathcal{V} = \frac{\gamma}{(1-\epsilon\lambda'(y)\rho'(\bar{x}))^2} + O\left(\frac{1}{1-\epsilon\lambda'(y)\rho'(\bar{x})}\right),$$

where

$$\begin{aligned}\gamma =& \epsilon^2\lambda'(y)^2(\rho(\bar{x})^2 - \rho(\bar{x}^2) + \rho'(\bar{x})(1-2x\rho(\bar{x})) - \bar{x}^2\rho'(\bar{x}^2)) + \\ & \epsilon^2\lambda'(y)^2\rho'(\bar{x})^2(\epsilon^2\lambda(y)^2 - \epsilon^2\lambda(y^2) - y^2\epsilon^2\lambda'(y^2)),\end{aligned}$$

with y the solution of $r_1(y) = 0$ for this particular channel parameter, $x = \epsilon\lambda(y)$ and $\bar{x} = 1 - x$.

Relation Between γ and α

Think of a decoder operating above the threshold of the code. Then, for large enough block-lengths, it will always get stuck before correcting all nodes. In Fig 2.5 we show the number of degree-one check nodes in the decoding as a function of the number of erased messages in the corresponding BP decoder. The quantity $\mathcal{V}$ represents the normalized variance of the number of erased messages in the decoder after an infinite number of iterations. In other words, the variance of the point at which the decoding trajectories hit the $R_1 = 0$ axis. This variance can be related to the variance of the number of degree one check nodes through the slope of the density evolution curve in the following way. Normalize all the quantities by $n\Lambda'(1)$ the number of edges in the graph. Consider the virtual curve $r_1(\epsilon, x)$ given by density evolution, and representing the fraction of degree-one check nodes in the residual graph, around the critical point for an erasure

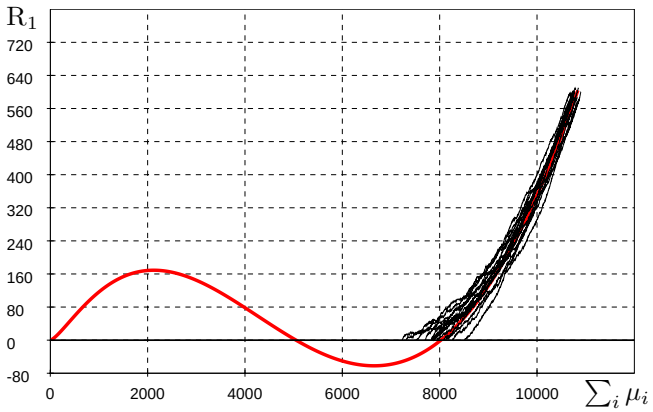


Figure 2.5: Number of degree one check nodes as a function of the number of erased messages in the corresponding BP decoder for LDPC($n = 8192, \lambda(x) = x^2, \rho(x) = x^5$). The thin lines represent the decoding trajectories that stop when $R_1 = 0$ and the thick line is the mean curve predicted by density evolution.

probability above the threshold (see Fig.2.6). The real decoding process stops when hitting the $R_1 = 0$ axis. Think then of a virtual process identical to the decoding for $R_1 > 0$ but that continues below the $R_1 = 0$ axis (see Appendix C). A simple calculation shows that if the point

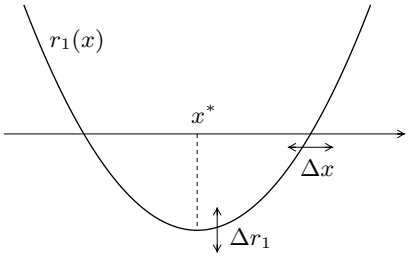


Figure 2.6: Virtual curve of $r_1(\epsilon, x)$ around the critical point for ϵ above the threshold.

at which the curve hits the x-axis varies by Δx while keeping the minimum at x^*, it results in a variation of the height of the curve by

$$\Delta r_1 = \frac{\partial^2 r_1(\epsilon, x)}{\partial x^2}\Big|_* (x - x^*)\Delta x + o(x - x^*)$$

Taking the expectation of the square on both side and letting ϵ tend to ϵ^*, we obtain

$$\begin{aligned}\delta^{r_1,r_1}|_* &= \lim_{\epsilon\to\epsilon^*}\left(\left(\frac{\partial^2 r_1(\epsilon,x)}{\partial x^2}\Big|_*\right)^2(x-x^*)^2\mathcal{V}+o((x-x^*)^2)\right)\\ &= \left(\frac{x^*}{\epsilon^*\lambda'(y^*)}\right)^2\lim_{\epsilon\to\epsilon^*}(1-\epsilon\lambda'(y)\rho'(\bar{x}))^2\mathcal{V}\\ &= \left(\frac{x^*}{\epsilon^*\lambda'(y^*)}\right)^2\gamma,\end{aligned}$$

with $\mathcal{V}$ being the variance introduced in the beginning of this section, and γ its dominant factor. The transition between the first and the second line comes simply from the study when ϵ tends to ϵ^*, of the relationship between ϵ and x, which are the solution of $r_1(\epsilon,x)=0$.

We conclude that the scaling parameter α can be obtained as

$$\begin{aligned}\alpha =&\sqrt{\frac{\delta^{r_1r_1}|_*}{\Lambda'(1)\left(\frac{\partial r_1}{\partial\epsilon}\right)^2}}=\sqrt{\frac{\gamma}{\Lambda'(1)x^{*2}\lambda'(y^*)^2\rho'(\bar{x}^*)^2}}\\ =&\left(\frac{\rho(\bar{x}^*)^2-\rho(\bar{x}^{*2})+\rho'(\bar{x}^*)(1-2x^*\rho(\bar{x}^*))-\bar{x}^{*2}\rho'(\bar{x}^{*2})}{\Lambda'(1)\lambda(y^*)^2\rho'(\bar{x}^*)^2}+\right.\\ &\left.\frac{\epsilon^2\lambda(y^*)^2-\epsilon^{*2}\lambda(y^{*2})-y^{*2}\epsilon^{*2}\lambda'(y^{*2})}{\Lambda'(1)\lambda(y^*)^2}\right)^{1/2}.\end{aligned}$$

2.5 Refined Scaling Law

In this section we explain in greater detail the arguments for Conjecture 1, and the procedure for computing the shift parameter β stated in Conjecture 2. As in the previous section, we shall first discuss this issue in an abstract setting in Section 2.5.1. The general procedure will then be applied to LDPC ensembles in Section 2.5.2.

2.5.1 The General Approach

Let us reconsider the setting of Section 2.4.1, i.e., a family of Markov chains $X_{n,0}, X_{n,1}, \ldots, X_{n,t}, \ldots$ taking values in $\mathbb{Z}^{d+1}$ and parametrized by the (large) integer n. As before we will drop in the sequel the subscript n to mitigate the notational burden. Throughout this section we shall assume the hypotheses of Lemma 4 to be fulfilled. Unlike in Section 2.4.1, we are interested in paths $X_0^t \equiv \{X_0, X_1, \ldots, X_t\}$ which are confined to the 'half space':

$$\mathbb{H}_+ \equiv \{x=(x^{(0)},\ldots,x^{(d)})\in\mathbb{Z}^{d+1}:\ \ x^{(0)}>0\}\,. \tag{2.24}$$

We would like to estimate the 'survival' probability

$$P_t \equiv \mathrm{P}(X_0^t \subseteq \mathbb{H}_+)\,. \tag{2.25}$$

Notice that P_t depends implicitly on the initial condition $X_0 = x_0 \in \mathbb{H}_+$. The coordinate $X_t^{(0)}$ should be thought as (an abstraction of) the number R_1 of degree-one check nodes in the analysis of iterative decoding. The survival probability P_t is therefore the probability of not having encountered a stopping set after t steps of the decoding process. We are interested in a time window of length $O(n)$. Without loss of generality we may fix $\tau_{\max} > 0$ and consider $t \in \{0, \dots, t_{\max}\}$ with $t_{\max} = \lfloor n\tau_{\max} \rfloor$.

We shall denote by $\overline{z}(\tau)$ the 'critical trajectory', i.e. a solution of the density evolution equations (2.15), such that $\overline{z}^{(0)}(\tau^*) = 0$, and $\overline{z}^{(0)}(\tau) > 0$ for any $\tau \in [0, \tau_{\max}]$, $\tau \neq \tau^*$. We call $z_0 = \overline{z}(0)$ the corresponding initial condition. In order to make contact with the application to iterative decoding, we shall make the following assumptions.

A. As $n \to \infty$, we have $x_0 = n\, z_0 + \sqrt{n}\, z_1 + O(1)$, with $z_1 \in \mathbb{R}^{d+1}$ independent of n. This corresponds to the erasure probability ϵ being in the critical window $\epsilon^* - \epsilon = O(n^{-1/2})$.

B. Let $\overline{z}_u(\tau)$, $u \in \mathbb{R}^{d+1}$, be a 'perturbed' critical trajectory obtained by solving the density evolution equations (2.15) with initial condition $\overline{z}_u(\tau^*) = \overline{z}(\tau^*) + u$. As for the critical trajectory, we consider this solution in the interval $[0, \tau_{\max}]$ and take u such that $|u| < \varepsilon$ with ε small enough. We assume that there exist a positive u-independent constant κ_1, and a function $u \mapsto a(u)$ such that

$$\overline{z}_u^{(0)}(\tau) - \overline{z}_u^{(0)}(\tau^*) \geq a(u)(\tau - \tau^*) + \kappa_1(\tau - \tau^*)^2$$

for any $\tau \in [0, \tau_{\max}]$.

C. We finally assume that $a(u)$ can be chosen in such a way that $|a(u)| < \kappa_2|u|$ for some positive constant κ_2.

Notice that the assumptions B and C above can be easily checked on the 'continuum' transition rates $\widehat{W}(\Delta|z)$ introduced in Section 2.4.1.

Consider the survival probability $P_{t_{\max}}$ at the 'latest' time. As we have seen in Section 2.4.1, most of the trajectories $X_0^{t_{\max}}$ are concentrated within $\sqrt{n}$ around $n\overline{z}(t/n)$. Therefore the absolute minimum of $X_t^{(0)}$ in the interval $\{0, \dots, t_{\max}\}$ will be realized for a t 'close' to $n\tau^*$. If

this absolute minimum is positive, the corresponding trajectory contributes to $P_{t_{\max}}$, otherwise it does not.

In order to formalize this argument, fix $t^* = \lfloor n\tau^* \rfloor$. Then

$$P_{t_{\max}} = \sum_{x \in \mathbb{H}_+} \mathrm{P}(X_0^{t_{\max}} \subseteq \mathbb{H}_+ | X_{t^*} = x)\ \mathrm{P}(X_{t^*} = x)\,. \tag{2.26}$$

Thanks to Lemma 4 we can accurately estimate the factor $\mathrm{P}(X_{t^*} = x)$. The term $\mathrm{P}(X_0^{t_{\max}} \subseteq \mathbb{H}_+ | X_{t^*} = x)$ is the probability that the global minimum of $X_t^{(0)}$, $t \in \{0 \dots t_{\max}\}$, is positive conditioned on $X_{t^*} = x$. Let us denote by t_{g} *a* 'time' for which the global minimum is realized.

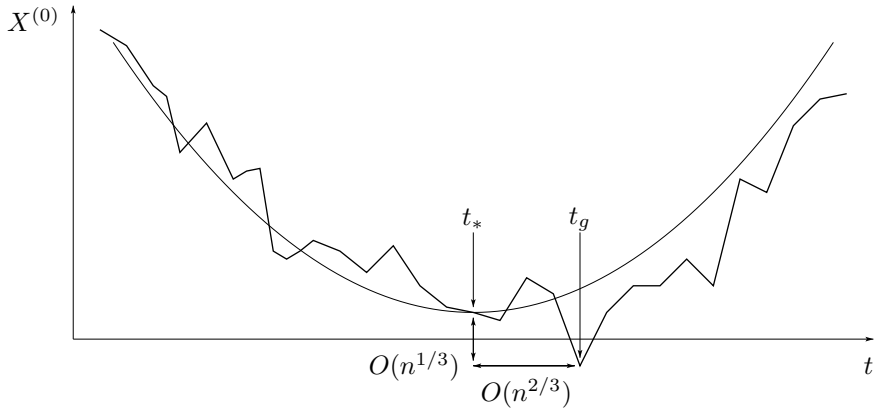


Figure 2.7: A pictorial view of decoding trajectories near the critical point. The type of trajectory depicted here is responsible for the shift appearing in the refined scaling form of Conjecture 1.

More precisely, $t_{\mathrm{g}} \in \{0 \dots t_{\max}\}$ is *a* random variable such that $X_{t_{\mathrm{g}}}^{(0)} \leq X_t^{(0)}$ for all $t \in \{0 \dots t_{\max}\}$. Call $\overline{z}_X(\tau)$ the perturbed critical trajectory defined above with perturbation vector $u = X_{t^*}/n - \overline{z}(\tau^*)$. In other words, we perturb the critical trajectory by an $O(1/\sqrt{n})$ amount in order to match it to the particular (finite n) realization of the Markov process we are dealing with within the critical region. Concentration arguments, analogous to the ones used to prove the point I of Lemma 4, imply that, for a given t:

$$\mathrm{P}\left\{ |X_t - n\overline{z}_X(t/n)| \geq \delta \sqrt{|t - t^*|} \right\} \leq \Omega_1\, e^{-\Omega_2\, \delta^2}\,,$$

for some positive constants Ω_1 and Ω_2 (as before we use this symbols to denote generic constants which are proven to exist independent of n). In fact a stronger condition holds true: by Doob's

maximal inequality [25, p. 227], for T fixed

$$\mathrm{P}\left\{\max_{|t-t^*|\leq T}\ |X_t - n\overline{z}_X(t/n)| \geq \delta\sqrt{T}\right\} \leq \Omega_1\, e^{-\Omega_2\,\delta^2}\,, \tag{2.27}$$

for some (possibly different) constants Ω_1 and Ω_2. Using this fact we can prove an useful result:

Lemma 5. Assume the same hypotheses as in Lemma 4 plus A, B and C above. Let $t_{\rm g}$ be a time at which the absolute minimum of $X_t^{(0)}$ is realized, for $t \in \{0 \dots t_{\max}\}$. Then there exist positive constants Ω_1, Ω_2 and δ_0, and a function $n_0(\delta)$ such that, for any $\delta > \delta_0$ and $n > n_0(\delta)$

$$\mathrm{P}\left\{|t_{\rm g} - t^*| \leq \delta^{2/3}\, n^{2/3},\ X_{t_g}^{(0)} \geq X_{t^*}^{(0)} - \delta^{4/3}\, n^{1/3}\right\} \geq 1 - \Omega_1\, \exp[-\Omega_2\, \delta^2]\,. \tag{2.28}$$

The proof is deferred to Appendix E. The content of this lemma is illustrated in Fig. 2.7.

The above result implies that corrections to the simplified scaling of Lemma 1 can be estimated through a two step procedure. In a nutshell: (i) Compute the probability for $X_{t^*}^{(0)}$ to be of order $n^{1/3}$; (ii) Evaluate the probability for $X_{t_{\rm g}}^{(0)}$ to be positive, conditioned on a given $X_{t^*}^{(0)}$ of order $n^{1/3}$.

Distribution of X_{t^*}

The simplified scaling form, cf. Lemma 1, was obtained by approximating the first factor in equation (2.26) by 1. The leading correction to this approximation comes from trajectories such that $X_{t^*}^{(0)} = O(n^{1/3})$. Because of Lemma 4, the probability distribution of $X_{t^*}^{(0)}$ (second factor) is well approximated by a Gaussian with center at $O(\sqrt{n})$ and variance of order n. The probability of having $X_{t^*}^{(0)} = O(n^{1/3})$ is therefore of order $n^{1/3} \cdot n^{-1/2} = n^{-1/6}$. This explains why the correction term in the refined scaling form (2.42) is of order $n^{-1/6}$.

This argument can be made more precise by rewriting equation (2.26) as

$$P_{t_{\max}} = \mathrm{P}(X_{t^*}^{(0)} > 0) - \sum_{x\in\mathbb{H}_+} \mathrm{P}(X_{t_{\rm g}}^{(0)} < 0 | X_{t^*} = x)\ \mathrm{P}(X_{t^*} = x)\,. \tag{2.29}$$

The first term corresponds to the simplified scaling form. We shall hereafter focus on the second one, $P_{\rm corr} \equiv \mathrm{P}(X_{t^*}^{(0)} > 0) - P_{t_{\max}}$. Notice that $\mathrm{P}(X_{t_{\rm g}}^{(0)} < 0 | X_{t^*} = x)$ varies much more rapidly (on a scale of order $n^{1/3}$) in $x^{(0)}$ than in the other coordinates (on a scale of order n). It is therefore useful to introduce the notation $\vec{x} = (x^{(1)} \dots x^{(d)})$ (and analogously $\vec{X}$ and $\vec{z}$) which distinguish explicitly the last d coordinates of x. Since $\mathrm{P}(X_{t^*} = x)$ varies on a scale $n^{1/2}$, we can safely approximate it by setting the coordinate $x^{(0)}$ to 0:

$$P_{\rm corr} = \sum_{\vec{x}} \left\{ \sum_{x^{(0)}>0} \mathrm{P}\left(X_{t_{\rm g}}^{(0)} < 0 | X_{t^*} = (x^{(0)}, \vec{x})\right) \right\} \mathrm{P}\left(X_{t^*} = (0, \vec{x})\right)\, (1 + O(n^{-1/6}))\,.$$

The term in curly brackets depends on $\vec{x}$ only through the transition coefficients in a neighborhood of $\vec{x}$ and varies therefore on a scale of order n. This point will be discussed in detail in the next section. On the contrary $\mathrm{P}\left(X_{t^*} = (0, \vec{x})\right)$ is peaked around $n\vec{z}(t^*/n)$ with a width of order $\sqrt{n}$. Therefore

$$P_{\rm corr} = \sum_{x^{(0)}>0} \mathrm{P}\left(X^{(0)}_{t_{\rm g}} < 0 | X_{t^*} = (x^{(0)}, n\vec{z}(\tau^*))\right) \mathrm{P}\left(X^{(0)}_{t^*} = 0\right) \left(1 + O(n^{-1/6})\right), \tag{2.30}$$

where we recall that $\vec{z}(\tau^*)$ denotes the last d coordinates of the critical point. The second factor can be evaluated easily using density and covariance evolution. Let us consider the application to iterative decoding (here $X^{(0)} \equiv \mathrm{R}_1$). Note that at the critical point and within the critical window $X^{(0)}$ is Gaussian with mean $\frac{\partial r_1}{\partial \epsilon}(\epsilon - \epsilon^*) n\Lambda'(1)$ and variance $n\Lambda'(1)\delta^{r_1, r_1}$. We therefore have

$$\mathrm{P}\left(X^{(0)}_{t^*} = 0\right) = \frac{1}{\Lambda'(1)\frac{\partial r_1}{\partial \epsilon}\sqrt{2\pi n\alpha^2}} \exp\left\{-\frac{n(\epsilon^* - \epsilon)^2}{2\alpha^2}\right\} \left(1 + O(n^{-1/2})\right).$$

This formula can indeed be guessed without any computation at all. The probability of $X^{(0)}_{t^*} = 0$ must be in fact proportional to the derivative of the probability of having $X^{(0)}_{t^*} \leq 0$, which is given by equation (2.23) within the critical window.

Distribution of the Global Minimum

We are left with the task of estimating the first factor in equation (2.30), and more generally the probability distribution of $X^{(0)}_{t_{\rm g}}$ conditioned on X_{t^*}. Lemma 5 is, once again, quite helpful. The difference $|t_{\rm g} - t^*|$ is small on the scale n on which the transition rates are state-dependent. This suggests that the leading correction to the simplified scaling depends on the transition rates only through their behavior at the critical point $\overline{z}(\tau^*)$. On the other hand, $|t_{\rm g} - t^*|$ is large on the scale $O(1)$ of a single step. We can therefore hope to compute the leading correction within a 'continuum' approach.

More precisely, define the rescaled trajectory $u(\cdot) \in \mathbb{R}^{d+1}$ by taking

$$u^{(0)}(n^{-2/3}(t - t^*)) \equiv n^{-1/3} X^{(0)}_t, \tag{2.31}$$

$$u^{(i)}(n^{-2/3}(t - t^*)) \equiv n^{-2/3}(X^{(i)}_t - X^{(i)}_{t^*})\, i = 1, \cdots, d, \tag{2.32}$$

for integers t such that $|t - t^*| \leq \theta_{\rm MAX} n^{2/3}$, and interpolating linearly among these points. A textbook result in the theory of stochastic processes [26] implies the following lemma.

Lemma 6. Let X be distributed as above under the condition $X_{t^*} = (n^{1/3}\zeta, n\vec{z}(\tau^*))$. The process $u(\cdot)$ defined in equations (2.31) and (2.32) converges as $n \to \infty$ to a diffusion process with generator:

$$\mathcal{L}_d = -\left(\sum_{i=1}^{d} \omega_i^* u^{(i)}\right) \frac{\partial}{\partial u^{(0)}} - \sum_{i=1}^{d} f_*^{(i)} \frac{\partial}{\partial u^{(i)}} + \frac{1}{2} f_*^{(00)} \frac{\partial^2}{\partial (u^{(0)})^2}\,, \tag{2.33}$$

conditioned on $u^{(0)}(0) = \zeta$, and $\vec{u}(0) = \vec{0}$. In the above formula we used the notation

$$f_*^{(i)} = \hat{f}^{(i)}(\overline{z}(\tau^*)), \qquad f_*^{(ij)} = \hat{f}^{(ij)}(\overline{z}(\tau^*)), \quad \omega_i^* = \left.\frac{\partial \hat{f}^{(0)}}{\partial z_i}\right|_{\overline{z}(\tau^*)}.$$

In order not to burden the presentation, the proof of this statement is postponed to Appendix F. Notice that the only role of $\theta_{\rm MAX}$ in the above lemma is to assure that $u(\theta)$ stays within a finite neighborhood of $u(0)$ with high probability. We want to use the process $u(\theta)$ in order to compute the second factor in equation (2.30) and therefore the distribution of the absolute minimum of $u(\theta)$. Let us call $\theta_{\rm g}$ the location of the minimum. Lemma 5 implies that $|\theta_{\rm g}| < \delta^{4/3}$ with probability at least $1 - \Omega_1 \exp(-\Omega_2 \delta^2)$. We can therefore safely let $\theta_{\rm MAX} \to \infty$ and consider the diffusion process defined above for $\theta \in (-\infty, +\infty)$.

Notice that only the first derivative with respect to the coordinates $u^{(1)}, \dots, u^{(d)}$ appears in equation (2.33). The process $\vec{u}(\theta)$ is therefore deterministic: $u^{(i)}(\theta) = f_*^{(i)}\theta$ for $i = 1, \dots, d$. We can substitute this behavior in equation (2.33) and deduce that $u^{(0)}(\theta)$ is a time-dependent diffusion process with generator

$$\mathcal{L}_0(\theta) = -\left(\sum_{i=1}^{d} \omega_i^* f_*^{(i)}\right) \theta\, \frac{\partial}{\partial u^{(0)}} + \frac{1}{2} f_*^{(00)} \frac{\partial}{\partial (u^{(0)})^2}. \tag{2.34}$$

It is convenient to rescale $u^{(0)}$ and θ in order to reduce the above generator to a standard form:

$$\overline{\theta} = (f_*^{(00)})^{-1/3} \left(\sum_{i=1}^{d} \omega_i^* f_*^{(i)}\right)^{2/3} \theta\,, \qquad w = (f_*^{(00)})^{-2/3} \left(\sum_{i=1}^{d} \omega_i^* f_*^{(i)}\right)^{1/3} u^{(0)}\,. \tag{2.35}$$

The generator for $w(\overline{\theta})$ has now the form (we keep the same name with an abuse of notation)

$$\mathcal{L}_0(\bar{\theta}) = -\bar{\theta}\, \frac{\partial}{\partial w} + \frac{1}{2} \frac{\partial}{\partial w^2}\,. \tag{2.36}$$

A little thought shows that this is equivalent to saying that $w(\overline{\theta}) = w(0) + \overline{\theta}^2/2 + B(\overline{\theta})$ with $B(\overline{\theta})$ a two-sided standard Brownian motion with $B(0) = 0$. The problem of computing the distribution of the global minimum of such a process has been solved in [27]. Adapting the results of this paper we find

$$\mathrm{P}\left(w(\overline{\theta}_{\rm g}) - w(0) < -z\right) = 1 - K(z)^2\,, \tag{2.37}$$

where

$$K(z) = \frac{1}{2} \int \frac{\mathrm{Ai}(iy)\mathrm{Bi}(2^{1/3}z + iy) - \mathrm{Ai}(2^{1/3}z + iy)\mathrm{Bi}(iy)}{\mathrm{Ai}(iy)} \, dy \,. \tag{2.38}$$

with $\mathrm{Ai}(\cdot)$ and $\mathrm{Bi}(\cdot)$ the Airy functions defined in [28].

Putting everything together we get

$$\begin{aligned} \sum_{x^{(0)}>0} \mathrm{P}\left(X_{t_{\mathrm{g}}}^{(0)} < 0 | X_{t^*} = (x^{(0)}, n\vec{z}(t^*/n))\right) &= n^{1/3}\Omega \, (f_*^{(00)})^{2/3} \left(\sum_{i=1}^{d} \omega_i^* f_*^{(i)}\right)^{-1/3} (1 + o(1)) \,, \\ &= n^{1/3}\Omega\beta_0(1 + o(1)) \end{aligned}$$

with

$$\Omega \equiv \quad \int_0^\infty [1 - K(z)^2] \, dz \tag{2.39}$$

$$\beta_0 \equiv \quad (f_*^{(00)})^{2/3} \left(\sum_{i=1}^{d} \omega_i^* f_*^{(i)}\right)^{-1/3} \tag{2.40}$$

A numerical computation yields $\Omega = 1.00(1)$.

Finally, we can write

$$P_{\mathrm{corr}} = \frac{n^{1/3}\Omega\beta_0}{\Lambda'(1)\frac{\partial r_1}{\partial \epsilon}\sqrt{2\pi n\alpha^2}} \exp\left\{-\frac{n(\epsilon^* - \epsilon)^2}{2\alpha^2}\right\} (1 + O(n^{-1/6})) \tag{2.41}$$

2.5.2 Application to Low-Density Parity-Check Ensembles

There is one important difficulty in applying the general scheme explained above to iterative decoding: the Markov process is not defined for $\mathrm{R}_1 < 0$. Recall that R_1 corresponds, in this context, to the 'critical' variable $X_t^{(0)}$. On the other hand, both the drift and diffusion coefficients $\hat{f}^{(i)}(\cdot)$ and $\hat{f}^{(ij)}(\cdot)$ can be continued analytically through the $\mathrm{R}_1 = 0$ plane. Since the final result (2.42) depends on the transition rates only through these quantities, we are quite confident that it remains correct also for iterative decoding applications.

Now combining all expressions in the case of decoding gives us

$$\begin{aligned}
\mathrm{P}_{\mathrm{B},\gamma}(n,\lambda,\rho,\epsilon) &= \left(Q\left(\frac{\sqrt{n}(\epsilon^*-\epsilon)}{\alpha}\right)+P_{\mathrm{corr}}\right)(1+o_n(n^{-1/3})),\\
&= \left(Q\left(\frac{\sqrt{n}(\epsilon^*-\epsilon)}{\alpha}\right)+\frac{n^{1/3}\Omega\beta_0}{\Lambda'(1)\frac{\partial r_1}{\partial\epsilon}\sqrt{2\pi n\alpha^2}}\exp\left\{-\frac{n(\epsilon^*-\epsilon)^2}{2\alpha^2}\right\}\right)(1+o_n(n^{-1/3})),\\
&= \left(Q\left(\frac{\sqrt{n}(\epsilon^*-\epsilon)}{\alpha}\right)+\beta n^{-1/6}\frac{1}{\sqrt{2\pi\alpha^2}}\exp\left\{-\frac{n(\epsilon^*-\epsilon)^2}{2\alpha^2}\right\}\right)(1+o_n(n^{-1/3})),
\end{aligned} \tag{2.42}$$

$$\begin{aligned}
&= Q\left(\frac{\sqrt{n}(\epsilon^*-\beta n^{-\frac{2}{3}}-\epsilon)}{\alpha}\right)(1+o_n(n^{-1/3})),\\
&= Q\left(\frac{z}{\alpha}\right)(1+o_n(n^{-1/3})),
\end{aligned}$$

where $z=\sqrt{n}(\epsilon^*-\beta n^{-\frac{2}{3}}-\epsilon)$ is kept fixed while n increases. The value of β is obtained from (2.40) and (2.42)

$$\beta/\Omega \;=\; -\left(\frac{f^{(r_1r_1)}}{\Lambda'(1)}\right)^{2/3}\left[\sum_{i=2}^{\mathtt{l}_{\max}} f^{(l_i)}\frac{\partial f^{(r_1)}}{\partial l_i}+\sum_{i=2}^{\mathtt{r}_{\max}-1} f^{(r_i)}\frac{\partial f^{(r_1)}}{\partial r_i}\right]^{-1/3}\left(\frac{\partial r_1}{\partial\epsilon}\right)^{-1}. \tag{2.43}$$

Then, replacing the drifts and covariances by their values that we computed in Section 2.4.2 and dropping Ω which numerical value is 1.00 independently of the code, we obtain the formula expressed in Conjecture 2

$$\beta = \left(\frac{\epsilon^{*4}r_2^{*2}(\epsilon^*\lambda'(y^*)^2r_2^*-x^*(\lambda''(y^*)r_2^*+\lambda'(y^*)x^*))^2}{\Lambda'(1)^2\rho'(\bar{x}^*)^3x^{*10}(2\epsilon^*\lambda'(y^*)^2r_3^*-\lambda''(y^*)r_2^*x^*)}\right)^{1/3}, \tag{2.44}$$

with for $i\geq 2$

$$r_i^*=\sum_{m\geq j\geq i}(-1)^{i+j}\binom{j-1}{i-1}\binom{m-1}{j-1}\rho_m(\epsilon^*\lambda(y^*))^j.$$

For regular ensembles $\lambda(x)=x^{\mathtt{l}-1}$ and $\rho(x)=x^{\mathtt{r}-1}$ these expressions simplify to

$$\beta \;=\epsilon^*\left(\tfrac{\mathtt{l}-2}{\mathtt{l}x^*y^*}\right)^{2/3}\left(\tfrac{1}{(\mathtt{l}-1)}+\tfrac{(\mathtt{r}-2)x^*}{1-x^*}-2\right)^{-1/3}. \tag{2.45}$$

2.6 Approximation of the Waterfall Curve

Consider the ensemble LDPC$(n,\lambda(x),\rho(x))$, transmission over the BEC of erasure probability ϵ. The formulas provided in Lemma 1 and Conjecture 1 can be used to find approximations of the performance of the codes. However, it is fair to ask how good these approximations are.

The scaling law that we prove in this chapter tells us that the error probability due to large error events converges to the formulas in Lemma 1. But, this convergence is such for $z = \sqrt{n}(\epsilon^* - \epsilon)$ kept fixed while n tends to $+\infty$. The situation is identical for Conjecture 1. This indicates that the approximation resulting from the scaling law should be accurate in a window of width $1/\sqrt{n}$ around the threshold. Another issue is the speed of convergence of the resulting approximation. We conjecture it be at least $O(n^{-1/3})$, which is still quite slow.

Fortunately, in practice, one observes that the approximations are accurate over a wide range and already for short blocklengths. In Fig. 2.8, we compare the exact block error probability due to large failures to our approximation for the ensemble $\lambda(x) = x^2$ and $\rho(x) = x^5$ for several blocklengths. In Fig. 2.9, we do the same comparison, but this time for an irregular degree

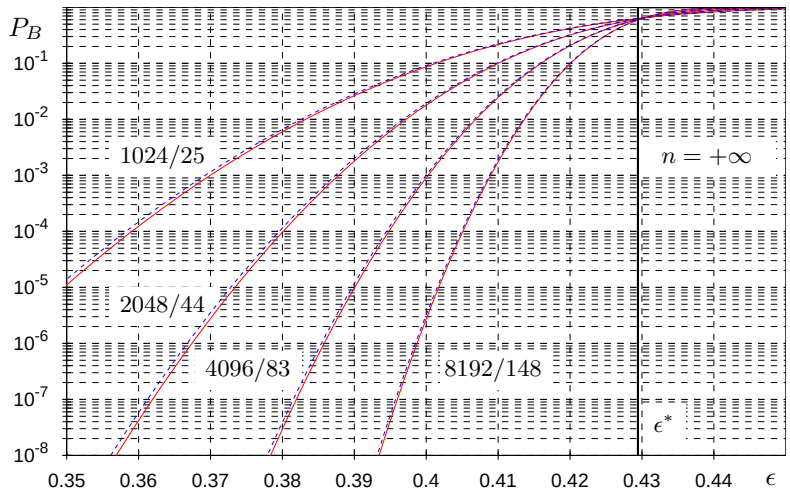


Figure 2.8: Block error probability curves for LDPC$(n, \lambda(x) = x^2, \rho(x) = x^5)$ when used over a binary erasure channel of erasure fraction ϵ. The different solid curves are the exact block error probabilities for $n \in \{1024, 2048, 4096\}$ where we only count error due to stopping set of sizes bigger or equal to $\{25, 44, 83, 148\}$ respectively. The threshold is $\epsilon^* = 0.4294381$. The dashed curves are obtained through our scaling law with $\alpha = 0.56035834$ and $\beta = 0.61694874$.

distribution $\lambda(x) = 1/6x + 5/6x^3$ and $\rho(x) = x^5$ and for even shorter blocklengths. We can see that even for length $n = 350$, the curves are quite close. As a consequence, it is clear that this approximation can be used to predict the performance of the codes for short and moderate blocklength.

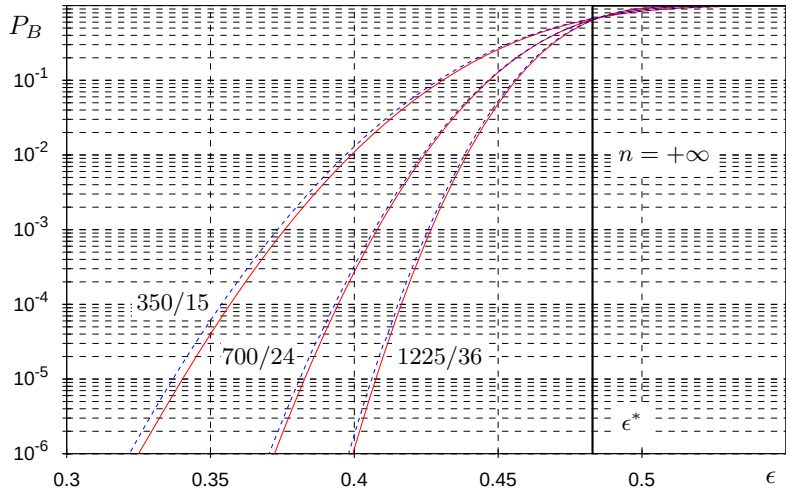


Figure 2.9: Block error probability curves for LDPC$(n, \lambda(x) = 1/6x + 5/6x^3, \rho(x) = x^5)$ when used over a binary erasure channel of erasure fraction ϵ. The different solid curves are the exact block error probabilities for $n \in \{350, 700, 1225\}$ where we only count error due to stopping set of sizes bigger or equal to $\{15, 24, 36\}$ respectively. The threshold is $\epsilon^* = 0.48280278$. The dashed curves are obtained through our scaling law with $\alpha = 0.57911955$ and $\beta = 0.68869134$.

Simple Approximation

Let us outline at this point, the different steps required to compute this approximation. Assume we want to approximate the block and bit error probability curves in the waterfall region for the ensemble LDPC$(n, \lambda(x), \rho(x))$ and for several blocklengths. We call the resulting quantities respectively, $\mathrm{P_B^W}(n, \lambda(x), \rho(x), \epsilon)$ and $\mathrm{P_b^W}(n, \lambda(x), \rho(x), \epsilon)$.

1. Use density evolution to find the threshold of the code and the whole set of critical values $(\epsilon^*, y^*, x^*, \nu^*)$.

2. Compute α and β using the formulas

$$\alpha = \left(\frac{\rho(\bar{x}^*)^2 - \rho(\bar{x}^{*2}) + \rho'(\bar{x}^*)(1 - 2x^*\rho(\bar{x}^*)) - \bar{x}^{*2}\rho'(\bar{x}^{*2})}{\Lambda'(1)\lambda(y^*)^2\rho'(\bar{x}^*)^2} + \frac{\epsilon^{*2}\lambda(y^*)^2 - \epsilon^{*2}\lambda(y^{*2}) - y^{*2}\epsilon^{*2}\lambda'(y^{*2})}{\Lambda'(1)\lambda(y^*)^2} \right)^{1/2},$$

$$\beta = \left(\frac{\epsilon^{*4} r_2^{*2} (\epsilon^* \lambda'(y^*)^2 r_2^* - x^*(\lambda''(y^*) r_2^* + \lambda'(y^*) x^*))^2}{\Lambda'(1)^2 \rho'(\bar{x}^*)^3 x^{*10} (2\epsilon^* \lambda'(y^*)^2 r_3^* - \lambda''(y^*) r_2^* x^*)} \right)^{1/3}$$

with for $i \geq 2$

$$r_i^* = \sum_{m \geq j \geq i} (-1)^{i+j} \binom{j-1}{i-1} \binom{m-1}{j-1} \rho_m (\epsilon^* \lambda(y^*))^j.$$

3. Plug these values in the formula

$$\mathrm{P}_{\mathrm{B}}^{W}(n, \lambda(x), \rho(x), \epsilon) = Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right),$$
$$\mathrm{P}_{\mathrm{b}}^{W}(n, \lambda(x), \rho(x), \epsilon) = \nu^* Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right),$$

to obtain the approximation.

Multiple Critical Points

For short blocklengths and certain degree distributions, one can further improve the above approximation. Consider the ensemble LDPC$(n, \lambda(x), \rho(x))$ with
$\lambda(x) = 0.317226x^1 + 0.206346x^2 + 0.114107x^3 + 0.0236055x^4 + 0.00306835x^5 + 0.109719x^{11} + 0.225928x^{12}$
and $\rho(x) = 0.0516065x^4 + 0.64271x^5 + 0.305684x^6$. The threshold of the code is $\epsilon^* = 0.54749662$ and the set of critical values $(\epsilon^* = 0.54749662, y^* = 0.56360173, x^* = 0.14659268, \nu^* = 0.12537947)$.

On Fig. 2.10, we plot $r_1(y)$ for several values of ϵ at and close to the threshold. Our scaling law tells us, that the contribution stemming from trajectories that die at or around the critical point $(y^* = 0.56360173)$ will dominate in the error probability as the blocklength increases. However, from the shape of the curve, we see that some trajectories might die before as the curve gets narrower around $y = 0.91$. This contribution that will become negligible as the blocklength increases will still affect short and moderate blocklengths. Fortunately, one can easily estimate this contribution.

Using the same arguments as beforehand, consider the decoding trajectories for $y > 0.8$. Their distribution is a Gaussian with a certain mean and covariance that we can estimate through covariance evolution. As before, we have a non-vanishing contribution to the error probability when the mean of the number of check nodes of degree 1, namely R_1 is of the order of $\sqrt{n}$ and for this case we can use a Q-function to estimate this probability.

The overall procedure to find our refined approximation becomes.

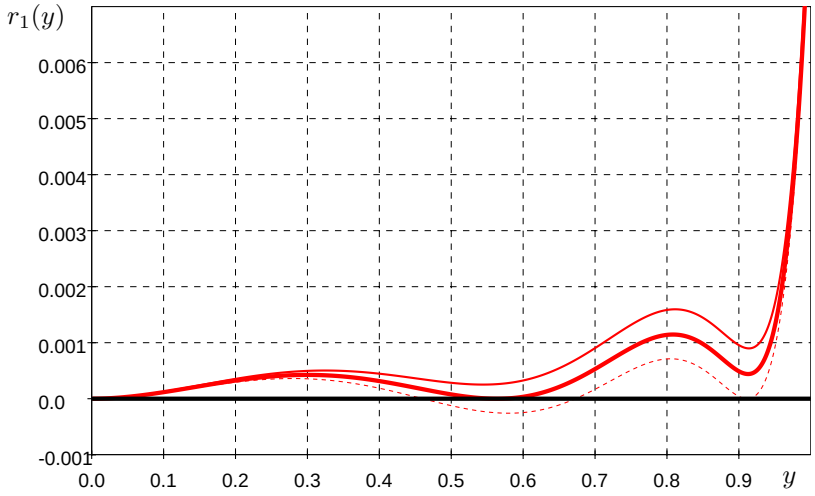


Figure 2.10: The asymptotic fraction $r_1(y)$ for $\lambda(x) = 0.317226x^1 + 0.206346x^2 + 0.114107x^3 + 0.0236055x^4 + 0.00306835x^5 + 0.109719x^{11} + 0.225928x^{12}$ and $\rho(x) = 0.0516065x^4 + 0.64271x^5 + 0.305684x^6$. The plain curve is for $\epsilon = 0.545$ and the thick curve is for $\epsilon^* = 0.54749662$ and the dashed curve for $\epsilon = 0.54990017$.

1. Use density evolution to find the threshold of the code as well as all critical points. In this case, we have two critical points.

   - The one corresponding to the threshold for $y = 0.56360173$
     $(\epsilon^* = 0.54749662, y^* = 0.56360173, x^* = 0.14659268, \nu^* = 0.12537947)$.

   - The critical point at $y = 0.91171366$, which is
     $(\tilde{\epsilon} = 0.54990017, \tilde{y} = 0.91171366, \tilde{x} = 0.37375033, \tilde{\nu} = 0.41133613)$.

2. Compute (α, β) and $(\tilde{\alpha}, \tilde{\beta})$ respectively for the two critical points.

3. Approximate the error probability by

$$\begin{aligned}
\mathrm{P}_{\mathrm{B}}^{W}(n,\lambda(x),\rho(x),\epsilon) =& Q\left(\frac{\sqrt{n}(\tilde{\epsilon}-\tilde{\beta}n^{-\frac{2}{3}}-\epsilon)}{\tilde{\alpha}}\right) \\
&+\left(1-Q\left(\frac{\sqrt{n}(\tilde{\epsilon}-\tilde{\beta}n^{-\frac{2}{3}}-\epsilon)}{\tilde{\alpha}}\right)\right)Q\left(\frac{\sqrt{n}(\epsilon^{*}-\beta n^{-\frac{2}{3}}-\epsilon)}{\alpha}\right), \\
\mathrm{P}_{\mathrm{b}}^{W}(n,\lambda(x),\rho(x),\epsilon) =& \tilde{\nu}Q\left(\frac{\sqrt{n}(\tilde{\epsilon}-\tilde{\beta}n^{-\frac{2}{3}}-\epsilon)}{\tilde{\alpha}}\right) \\
&+\left(1-Q\left(\frac{\sqrt{n}(\tilde{\epsilon}-\tilde{\beta}n^{-\frac{2}{3}}-\epsilon)}{\tilde{\alpha}}\right)\right)\nu^{*}Q\left(\frac{\sqrt{n}(\epsilon^{*}-\beta n^{-\frac{2}{3}}-\epsilon)}{\alpha}\right).
\end{aligned}$$

   In this approximation, the first term corresponds to the probability that trajectories die in the first critical point and the second terms is the probability that the trajectories die in the second critical point conditioned on surviving the first.

This procedure can be applied to any number of critical points.

Appendices

A - When the Threshold is Given by the Stability Condition

As already mentioned, ensembles whose threshold are given by the stability condition [20] are expected to follow a different scaling from the one described in Lemma 1. We will limit our discussion to the simplest case, namely the case of cycle code ensembles. We conjecture though that the form of the scaling law is quite general and applies to all ensembles whose thresholds are given by the stability condition. The cycle Poisson ensemble ELDPC$(n, \lambda(x) = x, r, s)$ is constructed in the following way. Define n variable nodes having degree two. Define $n(1-r)$ check nodes without assigning them any degree. To sample an element from the ensemble, connect each edge emanating from the variable nodes to a check node chosen at random with uniform probability. The rate of the ensemble is r and the degree distribution of the check nodes tends to $\rho(x) = e^{\frac{(x-1)}{(1-r)\int_0^1 \lambda(u)du}}$ as $n \to \infty$. Finally, we remove from the ensemble all graphs containing cycles involving s or less variable nodes.

The cycle Poisson ensemble is slightly easier to handle analytically than the standard ensemble. We will therefore formulate our results mainly for this case.

Lemma 7. [Scaling of Block Probability for Cycle Poisson Ensembles] Consider transmission over BEC of erasure probability ϵ using elements from ELDPC$(n, \lambda(x) = x, r, s)$. Then

$$\mathrm{P_B}(n, \lambda(x) = x, r, s, n\epsilon) = 1 - A(s) a n^{-1/6} f(b\, n^{1/3}(\epsilon - \epsilon^*)) \left(1 + O(n^{-1/3})\right) ,$$

where $a = \bar{r}^{-1/6}$, $b = \bar{r}^{-2/3}$, $A(s) = \exp\left\{\sum_{s'=1}^{s} \frac{1}{2s'}\right\}$, and

$$f(x) = \frac{\sqrt{2\pi}3^{2/3}}{2}\, e^{-\frac{4x^3}{3}} p(3^{2/3}x; 3/2, -1) .$$

Hereby, $p(u; \alpha, \beta)$ is a so called *stable density* with representation

$$p(u; \alpha, \beta) = \frac{1}{2\pi} \int e^{-itu} \exp\left\{-|t|^{\alpha} e^{-i\frac{\pi}{2}K(\alpha)\beta \mathrm{sign}(t)}\right\} dt,$$

and $K(\alpha) = 1 - \mid 1 - \alpha \mid$.

Proof. In principle one could arrive at the above result by proceeding in the same fashion as for unconditionally stable ensembles, i.e., one could employ the tools of density evolution and covariance evolution.

We will however use an entirely different approach. Note that there is a one-to-one correspondence between elements of ELDPC$(n, \lambda(x) = x, r, s = 2)$ and random graphs on $n\bar{r}$ nodes with exactly n edges, see [29, 20]. If $s = 2$, then double edges and cycles of length four are excluded from the Tanner graph. Therefore, each variable node connects two distinct check nodes and no two variable nodes connect the same pair. If we therefore identify each variable node (and the two edges that emanate from it) with one edge in an ordinary graph we get our desired correspondence. Further, the decoder will be successful if and only if this random graph is a *forest*, i.e., a collection of trees. Let $F(l, k)$ denote the number of forests on l labeled nodes and k components. Such a forest has $l - k$ edges and therefore it corresponds to a constellation on $v = l - k$ variable nodes. Since these variable nodes can be ordered arbitrarily it follows that there are $v!F(n\bar{r}, n\bar{r} - v)$ constellations on v variable nodes which do not contain stopping sets.

It remains to find the total number of constellations on v variable nodes which are compatible with the expurgation scheme. The desired result will then follow by diving these two quantities. Assume $s = 0$. Then the total number of constellations on v variable nodes is equal to $(n\bar{r})^{2v}$, since for each edge we can choose one of the $n\bar{r}$ check nodes. Let $n_s(\mathtt{G})$ denote the number of cycles of length $2s$ in a fixed portion of the bipartite graph $\mathtt{G}$ of size v. It is easy to verify (and is a well studied problem in random graphs) that $\mathbb{E}[n_s(\mathtt{G})] = \frac{1}{2s}\left(\frac{2v}{n\bar{r}}\right)^s (1 + O(1/v))$. Further it is known that for each fixed s the random variables $(n_1, \cdots n_s)$ are asymptotically (as n and v tend to infinity with a fixed ratio) independent and follow a Poisson distribution, [30]. Finally, for the Poisson ensemble we have $\epsilon^* = \frac{\bar{r}}{2}$ so that around the critical value $v = \epsilon^* n = \frac{n\bar{r}}{2}$ and $\frac{2v}{n\bar{r}} = 1$. It follows that *around the threshold* the total number of constellations which are compatible with the expurgation scheme behaves like

$$T(v \sim n\epsilon^*) = (n\bar{r})^{2v} e^{-\sum_{s'=1}^{s} \frac{1}{2s'}} (1 + O(1/v)) = (n\bar{r})^{2v}/A(s)(1 + O(1/v)).$$

From this the block error probability around the threshold follows immediately once $F(l, k)$ is known, namely, if we have exactly $n\epsilon$ erasure we obtain

$$\mathrm{P_B}(n, \lambda(x) = x, r, s, n\epsilon \sim n\epsilon^*) = 1 - A(s)\frac{(n\epsilon)!F(n\bar{r}, n\bar{r} - n\epsilon)}{(n\bar{r})^{2n\epsilon}}(1 + O(1/n)).$$

One of the most celebrated formulas in enumerative combinatorics states that there are l^{l-2} labeled trees on l nodes, [31]. Unfortunately there does not seem to exist an equally elementary

expression for the number of labeled forests. The situation is aggravated by the fact that we are interested in the region where the average number of edges per node is around one. Exactly around this region the graph goes through a phase transition and so the behavior of $F(l,k)$ is nontrivial even in the limit of large sizes. Fortunately, the asymptotic behavior has been determined by Britkov [32] and the result has been made accessible (to the English speaking audience) in the book by Kolchin [33]. Our result now follows by employing the asymptotic approximation stated in Theorem 1.4.4 in [33].[1] □

Note, that for the cycle case the maximum likelihood and the iterative decoder perform *identical in terms of block error probability.* This is true since in this case the condition of no stopping sets is equal to the condition that there are no cycles which in turns implies that there is no codeword. Note, however, that this is *no longer true once we look at the resulting bit erasure probability.*

We also note that as we want to get the scaling law for the a BEC of erasure probability ϵ and not of a fixed erasure fraction, we need to convolve the above curves with the Binomial with mean $n\epsilon$. However, on the scale $\epsilon^* - \epsilon = O(n^{-1/3})$, the effect of the channel fluctuations vanishes in the large blocklength limit. The leading correction to the scaling law (2.46) coming from the channel consists in the substitution

$$f(x) \to f(x) + \frac{\epsilon^*(1-\epsilon^*)}{(1-r)^{4/3}}\, f''(x)\, n^{-1/3} + O(n^{-1/2})\,. \tag{2.46}$$

The following lemma characterizes the corresponding limiting block error probability curve.

Lemma 8. [Asymptotic Block Erasure Probability Curve] Consider transmission over BEC of erasure probability ϵ using random elements from ELDPC$(n, \lambda(x) = x, r, s)$. Then

$$\lim_{n\to\infty} \mathrm{P_B}(n, \lambda(x) = x, r, s, n\epsilon) = 1 - \sqrt{1 - \frac{\epsilon}{\epsilon^*}}\, \exp\left\{ \sum_{s'=1}^{s} \frac{\left(\frac{\epsilon}{\epsilon^*}\right)^{s'}}{2s'} \right\}.$$

The corresponding asymptotic bit error probability curve under iterative decoding can be obtained through a standard density evolution analysis and it is given in parametric form by

$$\left(\frac{x}{\lambda(1-\rho(1-x))}, \frac{x\Lambda(1-\rho(1-x))}{\lambda(1-\rho(1-x))} \right),$$

where $x \in (x^*, 1]$ and x^* is the solution to the equation $\epsilon^*\lambda(1-\rho(1-x)) = x$. Figure 2.11 shows the resulting bit and block error curves for ELDPC$(n, \lambda(x) = x, r = \frac{1}{2}, s = 1)$.

[1] The reader is warned that there is a slight typo in Theorem 1.4.4 as stated in [33].

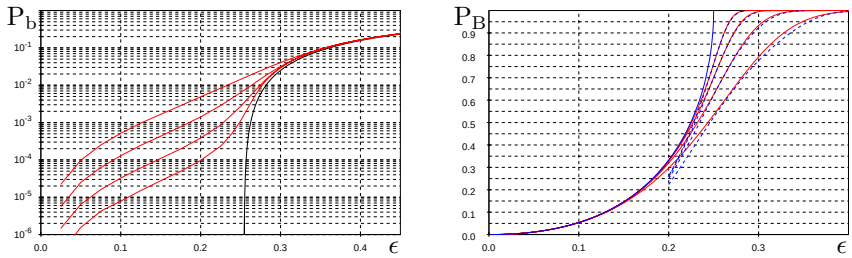


Figure 2.11: The bit and block erasure probability for ELDPC($n, \lambda(x) = x, r = \frac{1}{2}, s = 1$) for $n = 2^i$, $i = 8, 10, 12, 14$. As can be seen from the picture, the block erasure curves actually converge to a limiting (non-zero) curve over the whole range of ϵ, whereas the bit erasure curves decrease to zero below the threshold for increasing block lengths. Also shown are the result of using the scaling laws for the block erasure probability as stated in Lemma 7.

Cycle codes can not be expurgated up to some linear fraction of the blocklength since the number of stopping sets of size $s_1, \cdots s_k$ are jointly Poisson and have mean equal to $(2/\bar{r})^{s_i}/(2s_i)$, respectively. Below the threshold $\epsilon^* = \bar{r}/2$, the bit error probability scales as $1/n$. Expurgation changes uniquely the coefficient of this scaling. A simple calculation yields

$$\mathrm{P_b}(n, \lambda(x) = x, r, s, n\epsilon) = \frac{1}{2n} L_s\left(\frac{2\epsilon}{\bar{r}}\right)(1 + O(1/n))\,, \tag{2.47}$$

where we defined the function

$$L_s(x) := \sum_{s'=s+1}^{\infty} \frac{x^{s'}}{s'} = -\log(1-x) - \sum_{s'=1}^{s} \frac{x^{s'}}{s'}\,.$$

As shown in Fig. 2.12, this formula provides a good approximation to the bit error probability *away* from the critical region. Notice in fact that the coefficient of the $1/n$ term in Eq. (2.47) diverges as $\epsilon \to \epsilon^*$.

B - Covariance Evolution for a General Markov Process

In this Section we reconsider the abstract setting of section 2.16 and outline a proof of Lemma 4 under the assumptions 1-3.

Proof. We start with statement I, whose proof is fairly standard. Define a Doob's Martingale $\widehat{X}_0, \ldots, \widehat{X}_t$,

$$\widehat{X}_s = \mathbb{E}[X_t^{(i)} | X_0, \ldots X_s]\,.$$

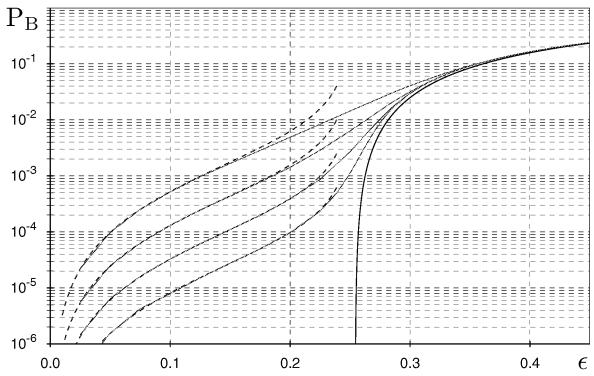


Figure 2.12: Comparison of the exact bit erasure curves (solid line) with the analytic expression given in (2.47) (dashed lines) for $n = 2^i$, $i = 8, 10, 12, 14$ and $\epsilon < \epsilon^*$.

Note that $\widehat{X}_t = X_t^{(i)}$ and $\widehat{X}_0 = \mathbb{E}[X_t^{(i)}] = \overline{X}_t^{(i)}$ so that

$$\mathrm{P}\{|X_t^{(i)} - \overline{X}_t^{(i)}| \geq \rho\} = \mathrm{P}\{|\widehat{X}_t - \widehat{X}_0| \geq \rho\}.$$

Therefore, by the the Hoeffding-Azuma [34, 35] inequality we will have proven (2.17) if we can show that $\widehat{X}_0, \ldots, \widehat{X}_t$ has bounded differences, more specifically, if we can show that

$$|\widehat{X}_s - \widehat{X}_{s-1}| \leq \sqrt{\Omega_0}, \;\; 1 \leq s \leq t.$$

To accomplish this task note that

$$|\widehat{X}_s - \widehat{X}_{s-1}| \leq \sup_{y,z} |\mathbb{E}[X_t^{(i)}|X_0 \ldots X_{s-1}, X_s = y] - \mathbb{E}[X_t^{(i)}|X_0 \ldots X_{s-1}, X_s = z]|\,, \tag{2.48}$$

where the sup is taken over all the y and z such that the trajectories $\{X_0, \ldots X_{s-1}, X_s = y\}$ and $\{X_0, \ldots X_{s-1}, X_s = z\}$ have non-vanishing probability. Consider therefore two realizations of the Markov chain which coincide up to time $s-1$ but are independent afterward. Denote them by $X_0, X_1, \ldots$ and $Y_0, Y_1, \ldots$, respectively, where by our assumption $X_\tau = Y_\tau$ for $0 \leq \tau \leq s-1$, but the processes evolve independently for $\tau \geq s$. Since by assumption $|X_s^{(i)} - X_{s-1}^{(i)}| \leq \kappa_1$ and $|Y_s^{(i)} - Y_{s-1}^{(i)}| \leq \kappa_1$ almost surely it follows that $|X_s^{(i)} - Y_s^{(i)}| \leq 2\kappa_1$ almost surely. Define $\delta X_\tau = X_\tau - Y_\tau$ and $\delta\overline{X}_\tau = \overline{X}_\tau - \overline{Y}_\tau$. Then we have for $s \leq \tau < t$

$$\begin{aligned} \delta\overline{X}_{\tau+1}^{(i)} &\leq \delta\overline{X}_\tau^{(i)} + \mathbb{E}[|f^{(i)}(X_\tau) - f^{(i)}(Y_\tau)|] \leq \\ &\leq \delta\overline{X}_\tau^{(i)} + \frac{A}{n}\delta\overline{X}_\tau^{(i)} + \frac{B}{n}\,. \end{aligned}$$

Here we approximated $f^{(i)}(X_\tau) - f^{(i)}(Y_\tau)$ by $\hat{f}^{(i)}(X_\tau/n) - \hat{f}^{(i)}(Y_\tau/n)$ and then used the fact that $\hat{f}^{(i)}(z)$ has bounded derivative. By Gronwall's Lemma we now get $|\overline{X}_t^{(i)} - \overline{Y}_t^{(i)}| < \sqrt{\Omega_0}$ for some suitable constant Ω_0. Since $\overline{X}_t^{(i)} = \mathbb{E}[X_t^{(i)}|X_0 \dots X_{s-1}, X_s = y]$ for some particular choice of y (and some fixed "past" $X_0 \dots X_{s-1}$) and the equivalent statement is true for $\overline{Y}_t^{(i)}$ it follows from (2.48) that $|\widehat{X}_s - \widehat{X}_{s-1}| \leq \sqrt{\Omega_0}$.

Notice that equation (2.17) implies

$$\mathbb{E}|X_t - \overline{X}_t|^p \leq \alpha_p (\Omega_0 t)^{p/2}\,, \tag{2.49}$$

for some[2] positive constants α_p. Before passing to the following parts of the Lemma, let us notice that not all the assumptions on the transition rates $\widehat{W}(\Delta|z)$ were used here. It is in fact sufficient to assume that the drifts $\hat{f}^{(i)}(z)$ are Lipschitz continuous.

Let us now consider the point II. A simple computation shows that

$$\begin{aligned}
\mathbb{E}\, X_{t+1}^{(i)} &= \mathbb{E}\, X_t^{(i)} + \mathbb{E}\, f^{(i)}(X_t)\,, & (2.50)\\
\mathbb{E}[X_{t+1}^{(i)}; X_{t+1}^{(j)}] &= \mathbb{E}[X_t^{(i)}; X_t^{(j)}] + \mathbb{E}\, f^{(ij)}(X_t) + & (2.51)\\
&\quad + \mathbb{E}[X_t^{(i)}; f^{(j)}(X_t)] + \mathbb{E}[f^{(i)}(X_t); X_t^{(j)}] + \mathbb{E}[f^{(i)}(X_t); f^{(j)}(X_t)]\,.
\end{aligned}$$

Consider the first of these equations and notice that, approximating $f^{(i)}(X_t)$ by $\hat{f}^{(i)}(X_t/n)$ one obtains

$$|\overline{X}_{t+1}^{(i)} - \overline{X}_t^{(i)} - \hat{f}^{(i)}(\overline{X}_t/n)| \leq \frac{A}{n} + |\mathbb{E}[\hat{f}^{(i)}(X_t/n) - \hat{f}^{(i)}(\overline{X}_t/n)]|\,. \tag{2.52}$$

Since the second derivative of $\hat{f}^{(i)}(z)$ is bounded, we have the estimate

$$\begin{aligned}
|\mathbb{E}[\hat{f}^{(i)}(X_t/n) - \hat{f}^{(i)}(\overline{X}_t/n)]| &\leq \left| \frac{1}{n} \sum_j \frac{\partial \hat{f}^{(i)}}{\partial z_j}\bigg|_{\overline{X}_t/n} \mathbb{E}[X_t^{(j)} - \overline{X}_t^{(j)}] \right| + \frac{B}{n^2}\mathbb{E}|X_t - \overline{X}_t|^2 \leq \\
&\leq \frac{C}{n}\,.
\end{aligned}$$

Summing equation (2.52) over t, and applying Gronwall's Lemma we get

$$\left| \frac{1}{n}\overline{X}_t^{(i)} - \overline{z}^{(i)}(t/n) \right| \leq \frac{A'}{n}\,. \tag{2.53}$$

Notice that if we limit ourself to assume Lipschitz continuous drift coefficients $\hat{f}^{(i)}(z)$, the same derivation yields a slightly weaker result: $|\overline{X}_t^{(i)}/n - \overline{z}^{(i)}(t/n)| \leq A'/\sqrt{n}$.

[2] One has in fact $\alpha_p = p\sqrt{\pi/2}\,\mathbb{E}|Z|^{p-1}$ with Z a standard Gaussian variable.

Equation (2.19) is proved from (2.51) much in the same way, the crucial input being an estimate on $\mathbb{E}|X_t - \overline{X}_t|^3$, once again obtained from equation (2.17). Here we limit ourselves to sketch how the various terms emerges. We start by rewriting equation (2.51) in the form

$$\begin{aligned}\Delta_{t+1}^{(ij)} &= \Delta_t^{(ij)} + \hat{f}^{(ij)}(\overline{X}_t/n) + \frac{1}{n}\sum_{l=1}^{d}\left[\Delta_t^{(il)}\left.\frac{\partial \hat{f}^{(j)}}{\partial z_l}\right|_{\overline{X}_t/n} + \left.\frac{\partial \hat{f}^{(i)}}{\partial z_l}\right|_{\overline{X}_t/n}\Delta_t^{(lj)}\right] + \\ &\quad + R_{ij}^{(0)} + R_{ij}^{(1)} + R_{ji}^{(1)} + R_{ij}^{(2)} + R_{ji}^{(2)} + R_{ij}^{(3)}\,,\end{aligned}$$

With the remainders listed below

$$\begin{aligned}R_{ij}^{(0)} &= \mathbb{E}[f^{(ij)}(X_t) - \hat{f}^{(ij)}(X_t/n)] + \mathbb{E}[\hat{f}^{(ij)}(X_t/n) - \hat{f}^{(ij)}(\overline{X}_t/n)]\,,\\ R_{ij}^{(1)} &= \mathbb{E}[X_t^{(i)}; f^{(j)}(X_t) - \hat{f}^{(j)}(X_t/n)]\,,\\ R_{ij}^{(2)} &= \mathbb{E}[X_t^{(i)}; \hat{f}^{(j)}(X_t/n) - \hat{f}^{(j)}(\overline{X}_t/n) - \frac{1}{n}\sum_{l=1}^{d}\left.\frac{\partial \hat{f}^{(j)}}{\partial z_l}\right|_{\overline{X}_t/n}(X_t^{(l)} - \overline{X}_t^{(l)})]\,,\\ R_{ij}^{(3)} &= \mathbb{E}[f^{(i)}(X_t); f^{(j)}(X_t)]\,.\end{aligned}$$

Each of this terms can be bounded separately as in the derivation of Eq. (2.53). Consider for instance $R_{ij}^{(1)}$:

$$\begin{aligned}|R_{ij}^{(1)}| &\le \mathbb{E}[X_t^{(i)}; X_t^{(i)}]^{1/2}\mathbb{E}[f^{(j)}(X_t) - \hat{f}^{(j)}(X_t/n); f^{(j)}(X_t) - \hat{f}^{(j)}(X_t/n)]^{1/2} \le\\ &\le An^{1/2}\frac{B}{n} \le \frac{C}{\sqrt{n}}\,,\end{aligned}$$

where we used the estimate (2.49).

Let us finally consider part III of the proposition, as stated in equation (2.21). It is easy to derive the following recursion for the generating function:

$$\begin{aligned}\Lambda_{t+1}(\lambda) &= \Lambda_t(\lambda) + \log\widetilde{W}(\lambda/\sqrt{n}|\overline{X}_t) - \frac{1}{\sqrt{n}}\lambda\cdot(X_t - \overline{X}_t) +\\ &\quad + \log\left\{\frac{\mathbb{E}[\widetilde{W}(\lambda/\sqrt{n}|X_t)\,e^{\frac{\lambda}{\sqrt{n}}\cdot X_t}]}{\mathbb{E}[\widetilde{W}(\lambda/\sqrt{n}|\overline{X}_t)\,e^{\frac{\lambda}{\sqrt{n}}\cdot X_t}]}\right\}\,.\end{aligned} \tag{2.54}$$

Here we defined the jump generating function

$$\widetilde{W}(\lambda|x) \equiv \sum_{\Delta} e^{\lambda\cdot\Delta}\; W(\Delta|x)\,.$$

The proof of equation (2.21) is completed by estimating the various terms in equation (2.54) as

follows

$$\left| \log \widetilde{W}(\lambda/\sqrt{n}|\overline{X}_t) - \frac{\lambda}{\sqrt{n}} \cdot (\overline{X}_{t+1} - \overline{X}_t) - \frac{1}{2n} \sum_{i,j} \hat{f}^{(ij)}(\overline{X}_t/n)\, \lambda_i \lambda_j \right| \;\leq\; \frac{\Omega_a(\lambda)}{n^{3/2}}\,,$$

$$\left| \frac{\mathbb{E}[(\widetilde{W}(\lambda/\sqrt{n}|X_t) - \widetilde{W}(\lambda/\sqrt{n}|\overline{X}_t))\, e^{\frac{\lambda}{\sqrt{n}}\cdot X_t}]}{\mathbb{E}[\widetilde{W}(\lambda/\sqrt{n}|\overline{X}_t)\, e^{\frac{\lambda}{\sqrt{n}}\cdot X_t}]} - \right.$$

$$\left. -\frac{1}{n^2} \sum_{l=1}^{d} \left[\frac{\partial \hat{f}^{(i)}}{\partial z_l}\bigg|_{\overline{X}_t/n} \Delta_t^{(lj)} + \Delta_t^{(il)}\, \frac{\partial \hat{f}^{(j)}}{\partial z_l}\bigg|_{\overline{X}_t/n} \right] \right| \;\leq\; \frac{\Omega_b(\lambda)}{n^{3/2}}\,.$$

We leave to the reader the pleasure of proving these two last (straightforward) inequalities. □

C - Extension of the Process and Proof of the Scaling Law

In this Appendix we prove Lemma 1. The idea is to regard iterative decoding as a Markov process in the space of states $\mathrm{x} = (\mathrm{R}_1, \cdots, \mathrm{R}_{\mathrm{r}_{\max}-1}, \mathrm{L}_1, \cdots, \mathrm{L}_{\mathrm{l}_{\max}}) \in \mathbb{Z}^d$. The transition rates and the initial condition for such a process are computed in Section 2.4.2. As in Section 2.4.1, we denote by $z = (r_1, \cdots, r_{\mathrm{r}_{\max}-1}, l_1, \cdots, l_{\mathrm{l}_{\max}})$ the normalized state and by $\overline{z}(\tau)$ the critical trajectory. This is the solution of the density evolution equations (2.15), such that $\overline{z}(\tau_{\mathrm{end}}) = (0,0,0)$, corresponding to complete decoding, $r_1(\tau^*) = 0$ for some $\tau^* \in (0, \tau_{\mathrm{end}})$, and $r_1(\tau) > 0$ for any $\tau \in (0, \tau_{\mathrm{end}})$, $\tau \neq \tau^*$.

It would be tempting to use the general covariance evolution approach provided by Lemma 4. However a simple remark prevents us from following this route in the most straightforward fashion. Lemma 4 was proved under the assumptions that the transition rates $\widehat{W}(\Delta|z)$ in the $n \to \infty$ limit become $C^2(\mathbb{R}^{d+1})$ functions of z. On the other hand, the decoding process is well defined only if $\mathrm{R}_1 > 0$, and we are interested in trajectories passing close to the 0 plane. In more concrete terms, Lemma 4 cannot be true when $\overline{z}(\tau)$ is at a distance of order $1/\sqrt{n}$ from the $r_1 = 0$ plane. The least that will happen is that a part of the Gaussian density is 'cut away'.

As a way to overcome this problem, we introduce a new Markov process on the same states $\mathrm{x} = (\mathrm{R}_1, \cdots, \mathrm{R}_{\mathrm{r}_{\max}-1}, \mathrm{L}_1, \cdots, \mathrm{L}_{\mathrm{l}_{\max}})$ which is well defined for $\mathrm{R}_1 \leq 0$. We extend the transition rates computed in Section 2.4.2 to $\mathrm{R}_1 \leq 0$ by setting $q_1 = 0$ in that case and as the q_i are probabilities, we set $q_{\mathrm{r}_{\max}} = 1 - \sum_{i=2}^{\mathrm{r}_{\max}-1} q_i$. More precisely this gives us the following transition

probabilities in the case where $R_1 \leq 0$

$$\begin{aligned} w_i(u_1, u_2, \cdots, u_{\mathbf{r}_{\max}}) &= 0, \qquad \text{if } u_1 \neq 1 \\ w_i(1, u_2, \cdots, u_{\mathbf{r}_{\max}}) &= \binom{i-1}{u_2, \cdots, u_{\mathbf{r}_{\max}}} q_2^{u_2} \cdots q_{\mathbf{r}_{\max}}^{u_{\mathbf{r}_{\max}}}. \end{aligned}$$

Such transition rates do not necessarily correspond to any graph process in the $\mathrm{R}_1 < 0$ plane. However, upon conditioning on $\mathrm{R}_1 > 0$ the 'extended' process coincides with the original one. Therefore the probability of not leaving the $\mathrm{R}_1 > 0$ half-space (the 'survival' probability) can be calculated on the extended process. Finally, let us notice that the precise form of this extension is immaterial as long as some requirements are met. Call $W(\Delta|x)$ the transition rates of the extended Markov process. We require that:

- The chain makes finite jumps.
- The rates are well approximated by their continuum counterpart $\widehat{W}(\Delta|z)$. As in Section 2.4.1 this means that $|W(\Delta|x) - \widehat{W}(\Delta|x/n)| \leq \kappa/n$.
- The continuum transition rates are C^2 with bounded derivatives in the region $\{\nu_{\mathtt{G}} > \varepsilon,\, r_1 > \varepsilon,\, \tau_{\mathtt{G}} > \varepsilon\}$ for any $\varepsilon > 0$.
- There exist a $\delta > 0$ such that the continuum drift coefficients are Lipschitz continuous uniformly in the region $\mathrm{Crit}(\delta) \equiv \{z \text{ s.t. } |z - \overline{z}(\tau^*)| < \delta\}$. This means that $|\hat{f}_i(z) - \hat{f}_i(z')| \leq \kappa' |z - z'|$ for some positive κ' and any pair of points $z, z' \in \mathrm{Crit}(\delta)$.

These requirements are easily checked on the extension defined above.

Recall from Lemma 1 that we are only interested in decoding errors of size at least $\gamma\nu^*$, where $\nu^* := \nu(\tau^*)$ is the fractional size of the graph at the critical point and γ is any number in $(0, 1)$. In particular γ is non-negative but can be chosen arbitrarily small. For ensembles with $\lambda'(0) = 0$ a simple union bound shows that the decoder will be successful with high probability once the residual graph is sufficiently small (see [20]) but if $\lambda'(0) > 0$ as we saw in Section 3, small small stopping sets can contribute non-negligibly to the error probability. Therefore, by choosing $\gamma \in (0, 1)$, we "separate out" the contributions to the block error probability which stem from large error events.

Fix $\tau_{\max}$ so that $\nu(\tau_{\max}) = \gamma\nu^*$. Define P_t to be the survival probability up to time t. It will be useful to denote by $P_t(x', t')$ the probability of surviving up to time t conditioned on having survived up to time t' and that the state at time t' is x'.

In order to apply Lemma 4 as far as we can, we decompose the time up to $t_{\max}$ into two intervals: $\{0, \dots, t^*_-\}$ and $\{t^*_- + 1, \dots, t_{\max}\}$. The survival probability can be written as

$$P_{t_{\max}} \;=\; \sum_x P_{t_{\max}}(x, t^*_-)\, P(x, t^*_-)\,. \tag{2.55}$$

Here $P(x,t)$ denotes the probability of arriving in state x' at time t' without hitting the $\mathrm{R}_1 = 0$ plane. The sum over x runs over the $\mathrm{R}_1 > 0$ half-space.

Next we chose $t^*_- = \lfloor n(\tau^* - \varepsilon) \rfloor$ for some (small) positive number ε. With this choice the factor $P(x, t^*_-)$ in the above equation can be estimated using the covariance evolution approach and Lemma 4. The reason is that the trajectories contributing to this factor stay at a distance of order n from the $\mathrm{R}_1 = 0$ apart from some exponentially rare cases. We leave to the reader the task of adapting the proof of Lemma 4.III to this situation.

The first factor in equation (2.55) can not be estimated through covariance evolution. Fortunately a less refined calculation is sufficient in this case. In fact the Lipschitz continuity of the drift coefficients ensures that, at any time $t > t^*_-$, the state is within δ of the density evolution prediction with probability at least $1 - \exp[-\delta^2/2\Omega(t - t^*_-)]$. This fact was stressed in the proof of Lemma 4, cf. Appendix B. For any state x, consider the solution $\overline{z}(\tau; x)$ of the density evolution equations (2.15) with initial condition $\overline{z}(t^*_-/n; x) = x/n$. Let $\widehat{P}_{t_{\max}}(x, t^*_-) = 0$ if $\overline{z}(\tau; x)$ intersects the $r_1 = 0$ plane in the interval $[t^*_-/n, \tau_{\max}]$ and $\widehat{P}_{t_{\max}}(x, t^*_-) = 1$ otherwise. The above concentration result implies that $\widehat{P}_{t_{\max}}(x, t^*_-)$ is a good approximation for $P_{t_{\max}}(x, t^*_-)$.

Let us prove the last statement in the cases in which $\overline{z}(\tau; x)$ does not intersect the $r_1 = 0$ plane (and therefore $\widehat{P}_{t_{\max}}(x, t^*_-) = 1$). If x is distributed according to $P(x, t^*_-)$, the trajectory $\overline{z}(\tau; x)$ will stay at a distance of order $1/\sqrt{n}$ from the critical one. In particular, its minimum distance from the $r_1 = 0$ plane will be $\gamma/\sqrt{n}$ with γ of order 1. This minimum will be achieved for τ close to τ_* with high probability. We therefore restrict ourselves to an interval of times $t^*_- < t < t^*_- + nT\varepsilon$ for some fixed number $T > 1$, and neglect the cases in which the r_1 plane is touched outside this interval. The error implied in substituting $\widehat{P}_{t_{\max}}(x, t^*_-)$ with $P_{t_{\max}}(x, t^*_-)$ is upper bounded by the probability that the maximum distance between the actual decoding trajectory and $\overline{z}(\tau; x)$ in the interval $t^*_- < t < t^*_- + nT\varepsilon$ ($\tau^* - \varepsilon < \tau < \tau_* + (T-1)\varepsilon$) is larger than $\gamma\sqrt{n}$. Using the above concentration result with $\delta = \gamma\sqrt{n}$ and $t - t^*_- < nT\varepsilon$, we get

$$|\widehat{P}_{t_{\max}}(x, t^*_-) - P_{t_{\max}}(x, t^*_-)| \le \exp\left\{-\frac{\gamma^2}{2\Omega T\varepsilon}\right\}. \tag{2.56}$$

As mentioned above, under the distribution $P(x, t^*_-)$, both γ and T are, with high probability

$O(1)$ (both with respect to $n \to \infty$ and $\varepsilon \to 0$). Therefore the right hand side of equation (2.56) can be made arbitrarily small by taking $\varepsilon \to 0$.

The last step consists in substituting $\widehat{P}_{t_{\max}}(x, t_-^*)$ for $P_{t_{\max}}(x, t_-^*)$ and the Gaussian density from covariance evolution for $P(x, t_-^*)$ in equation (2.55) and letting $n \to \infty$ with $n^{1/2}(\epsilon - \epsilon^*)$ fixed. This yields Lemma 1 up to corrections of which vanish when $\varepsilon \to 0$.

D - Variance of Erased Edges

Consider the ensemble LDPC(n, λ, ρ) and transmission over the BEC of erasure probability ϵ. As pointed out in Section 2.4.3, the scaling parameter α can be related to $\mathcal{V}$, the (normalized) variance, with respect to the choice of the graph and the channel realization, of the number of erased edge messages sent from the variable nodes in the infinite graph in the limit of an infinite number of iterations. We start by first considering the case of a finite number of iterations ℓ. In this case we define the variables x_i, y_i, $\bar{x}_i = 1 - x_i$ and $\bar{y}_i = 1 - y_i$ where $i \in \{0, \cdots, \ell\}$, such that at each iteration i of BP and for an infinite blocklength, x_i represents the fraction of erased messages sent from the variable to the check nodes and y_i is the fraction of erased messages sent from the check to the variable nodes. These values are computed using density evolution [20] starting with $y_0 = 1$ and applying the recursion $y_{i+1} = 1 - \rho(1 - x_i)$, with $x_i = \epsilon\lambda(y_i)$. Using these variables, we have the following characterization of $\mathcal{V}^{(\ell)}$, the (normalized) variance of the number of erased edge messages sent from the variable nodes in the infinite graph after ℓ iterations.

Lemma 9. Let $\mathtt{G}$ be chosen uniformly at random from LDPC(n, λ, ρ) and consider transmission over the BEC of erasure probability ϵ. Label the $n\Lambda'(1)$ edges of $\mathtt{G}$ in some fixed order by the elements of $\{1, \cdots, n\Lambda'(1)\}$. Assume that the receiver performs ℓ rounds of Belief Propagation decoding and let $\mu_i^{(\ell)}$ be equal to one if the message sent at the end of the ℓ-th iteration along

edge i (from a variable node to a check node) is an erasure, and zero otherwise. Then

$$\begin{aligned}
\mathcal{V}^{(\ell)} = \lim_{n\to\infty} \frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2] - \mathbb{E}[(\sum_i \mu_i^{(\ell)})]^2}{n\Lambda'(1)} \qquad (2.57)\\
=& x_\ell + x_\ell(1,0)\left(\sum_{j=0}^{\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-j+1)\mathtt{C}(\ell-j)\right)(1,0)^T && \text{edges in } \mathtt{T}_1\\
&+ x_\ell^2\rho'(1)\sum_{i=0}^{\ell-1}\lambda'(1)^i\rho'(1)^i && \text{edges in } \mathtt{T}_2\\
&+ x_\ell(1,0)\left(\sum_{j=1}^{2\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-j+1)\mathtt{C}(\ell-j)\right)(1,0)^T && \text{edges in } \mathtt{T}_3\\
&+ (1,0)\left(\sum_{j=0}^{\ell}\left(y_{\ell-j}U^\star(j,j) + (1-y_{\ell-j})U^0(j,j)\right)\right. && \text{edges in } \mathtt{T}_4\\
&\qquad\left. + \sum_{j=\ell+1}^{2\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)\left(y_{\ell-j}U^\star(j,\ell) + (1-y_{\ell-j})U^0(j,\ell)\right)\right)\\
&- x_\ell\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G}_\mathtt{T}'}(1)]\\
&+ \sum_{i=1}^{\ell} F_i\left(x_i\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G}_\mathtt{T}'}(1)] - \epsilon\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G}_\mathtt{T}'}(y_i)]\right)\\
&- \sum_{i=1}^{\ell} F_i\epsilon\lambda'(y_i)\left(\mathbb{E}[\mu_1^{(\ell)}C^{\mathtt{G}_\mathtt{T}'}(1)]\rho(\bar{x}_{i-1}) - \mathbb{E}[\mu_1^{(\ell)}C^{\mathtt{G}_\mathtt{T}'}(\bar{x}_{i-1})]\right)\\
&+ \sum_{i=1}^{\ell-1} F_i\left(x_\ell + (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(1)\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)\\
&\qquad\cdot(1,0)\mathtt{V}(i)\mathtt{C}(i-1)\cdots\mathtt{V}(i-\ell+1)\mathtt{C}(i-\ell)(1,0)^T\\
&- \sum_{i=1}^{\ell-1} F_i x_i\left(x_\ell + (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(1)\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)(\lambda'(1)\rho'(1))^\ell
\end{aligned}$$

where

$$\mathtt{V}(i) = \begin{pmatrix} \epsilon\lambda'(y_i) & 0 \\ \lambda'(1)-\epsilon\lambda'(y_i) & \lambda'(1)\end{pmatrix}, \quad \mathtt{C}(i) = \begin{pmatrix} \rho'(1) & \rho'(1)-\rho'(\bar{x}_i) \\ 0 & \rho'(\bar{x}_i)\end{pmatrix}, \quad i \geq 0, \qquad (2.58)$$

$$\mathtt{V}(i) = \begin{pmatrix} \lambda'(1) & 0 \\ 0 & \lambda'(1)\end{pmatrix}, \qquad \mathtt{C}(i) = \begin{pmatrix} \rho'(1) & 0 \\ 0 & \rho'(1)\end{pmatrix}, \qquad i < 0. \qquad (2.59)$$

Further, $U^\star(j,j)$, $U^\star(j,\ell)$, $U^0(j,j)$ and $U^0(j,\ell)$ are computed through the following recursion.

For $j \leq \ell$, set

$$\begin{aligned} U^\star(j,0) =& (y_{\ell-j}\epsilon\lambda'(y_\ell), (1-y_{\ell-j})\epsilon\lambda'(y_\ell))^T, \\ U^0(j,0) =& (0,0)^T, \end{aligned}$$

whereas for $j > \ell$, initialize by

$$\begin{aligned} U^\star(j,j-\ell) =& (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)(1,0)^T M_1(j,j-\ell)(y_{\ell-j},1-y_{\ell-j})^T \\ &+ (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)(0,1)^T M_2(j,j-\ell)(y_{\ell-j},1-y_{\ell-j})^T, \\ U^0(j,j-\ell) =& (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)(0,1)^T\mathtt{V}(0)(y_{\ell-j},1-y_{\ell-j})^T. \end{aligned}$$

The recursion is

$$\begin{aligned} U^\star(j,k) =& M_1(j,k)\left(\mathtt{C}(\ell-j+k-1)U^\star(j,k-1)\right) \\ &+ M_2(j,k)\left(N_1(j,k-1)U^\star(j,k-1) + N_2(j,k-1)U^0(j,k-1)\right), \\ U^0(j,k) =& \mathtt{V}(\ell-j+k)\left(N_1(j,k-1)U^\star(j,k-1) + N_2(j,k-1)U^0(j,k-1)\right), \end{aligned}$$

with

$$\begin{aligned} M_1(j,k) &= \begin{pmatrix} \epsilon\lambda'(y_{\max\{\ell-k,\ell-j+k\}}) & 0 \\ \mathbb{1}_{\{j<2k\}}\epsilon(\lambda'(y_{\ell-k}) - \lambda'(y_{\ell-j+k})) & \epsilon\lambda'(y_{\ell-k}) \end{pmatrix}, \\ M_2(j,k) &= \begin{pmatrix} \mathbb{1}_{\{j>2k\}}\epsilon(\lambda'(y_{\ell-j+k}) - \lambda'(y_{\ell-k})) & 0 \\ \lambda'(1) - \epsilon\lambda'(y_{\min\{\ell-k,\ell-j+k\}}) & \lambda'(1) - \epsilon\lambda'(y_{\ell-k}) \end{pmatrix}, \\ N_1(j,k) &= \begin{pmatrix} \rho'(1) - \rho'(\bar{x}_{\ell-k-1}) & \rho'(1) - \rho'(\bar{x}_{\max\{\ell-k-1,\ell-j+k\}}) \\ 0 & \mathbb{1}_{\{j\leq 2k\}}(\rho'(\bar{x}_{\ell-j+k}) - \rho'(\bar{x}_{\ell-k-1})) \end{pmatrix}, \\ N_2(j,k) &= \begin{pmatrix} \rho'(\bar{x}_{\ell-k-1}) & \mathbb{1}_{\{j>2k\}}(\rho'(\bar{x}_{\ell-k-1}) - \rho'(\bar{x}_{\ell-j+k})) \\ 0 & \rho'(\bar{x}_{\min\{\ell-k-1,\ell-j+k\}}) \end{pmatrix}. \end{aligned}$$

The terms F_i are equal to

$$F_i = \prod_{k=i+1}^{\ell} \epsilon\lambda'(y_k)\rho'(\bar{x}_{k-1}), \tag{2.60}$$

and finally

$$\begin{aligned} \mathbb{E}[\mu_1^{(\ell)} V^{\mathtt{G}_\mathtt{T}'}(\alpha)] =& \sum_{k=0}^{2\ell} (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-k+1)\mathtt{C}(\ell-k)A(\ell,k,\alpha) \\ &+ x_\ell(\alpha\lambda'(\alpha)+\lambda(\alpha))\rho'(1)\sum_{i=0}^{\ell-1}\rho'(1)^i\lambda'(1)^i, \end{aligned}$$

with $A(\ell,k,\alpha)$ equal to

$$\begin{cases} \begin{pmatrix} \epsilon\alpha y_{\ell-k}\lambda'(\alpha y_{\ell-k}) + \epsilon\lambda(\alpha y_{\ell-k}) \\ \alpha\lambda'(\alpha) + \lambda(\alpha) - \epsilon\alpha y_{\ell-k}\lambda'(\alpha y_{\ell-k}) - \epsilon\lambda(\alpha y_{\ell-k}) \end{pmatrix}, & k \le \ell \\ \begin{pmatrix} \alpha\lambda'(\alpha) + \lambda(\alpha) \\ 0 \end{pmatrix}, & k > \ell \end{cases}$$

and

$$\begin{aligned} \mathbb{E}[\mu_1^{(\ell)} C^{\mathtt{G}_\mathtt{T}'}(\alpha)] = \sum_{k=1}^{2\ell} &(1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-k+2)\mathtt{C}(\ell-k+1)\mathtt{V}(\ell-k+1) \\ &\cdot \begin{pmatrix} \alpha\rho'(\alpha) + \rho(\alpha) - \alpha(\bar{x}_{\ell-k})\rho'(\alpha\bar{x}_{\ell-k}) - \rho(\alpha\bar{x}_{\ell-k}) \\ \alpha\bar{x}_{\ell-k}\rho'(\alpha\bar{x}_{\ell-k}) + \rho(\alpha\bar{x}_{\ell-k}) \end{pmatrix} \\ &+ x_\ell(\alpha\rho'(\alpha) + \rho(\alpha)) \sum_{i=0}^{\ell-1} \rho'(1)^i \lambda'(1)^i. \end{aligned}$$

Proof. Expand $\mathcal{V}^{(\ell)}$ in (2.57) as

$$\begin{aligned} \mathcal{V}^{(\ell)} &= \lim_{n\to\infty} \frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2] - \mathbb{E}[\sum_i \mu_i^{(\ell)}]^2}{n\Lambda'(1)}, \\ &= \lim_{n\to\infty} \frac{\sum_j \left(\mathbb{E}[\mu_j^{(\ell)} \sum_i \mu_i^{(\ell)}] - \mathbb{E}[\mu_j^{(\ell)}]\mathbb{E}[\sum_i \mu_i^{(\ell)}]\right)}{n\Lambda'(1)}, \\ &= \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_i \mu_i^{(\ell)}] - \mathbb{E}[\mu_1^{(\ell)}]\mathbb{E}[\sum_i \mu_i^{(\ell)}], \\ &= \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_i \mu_i^{(\ell)}] - n\Lambda'(1)x_\ell^2. \end{aligned} \tag{2.61}$$

In the last step, we have used the fact that $x_\ell = \mathbb{E}[\mu_i^{(\ell)}]$ for any $i \in \{1,\cdots,n\Lambda'(1)\}$. Let us look more carefully at the first term of (2.61). We are performing a finite number of iterations ℓ, so each message sent along an edge has a finite computation tree. In other words, each message is computed based on finite number of received values and the computation tree corresponds to all the variable nodes associated to these received values as well as all the check nodes connecting them. In the case where two messages have intersecting computation trees (at least one value received from the channel is involved in the computation of both messages), their values are evidently correlated. We can write

$$\lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)} \sum_i \mu_i^{(\ell)}] - n\Lambda'(1)x_\ell^2\right) = \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}} \mu_i^{(\ell)}] + \lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}^\mathtt{c}} \mu_i^{(\ell)}] - n\Lambda'(1)x_\ell^2\right), \tag{2.62}$$

where $\mathtt{T}$ contains the indices of all edges, such that their computation tree intersects the computation tree of $\mu_1^{(\ell)}$. This means that these edges carry messages that are computed based on the same received values as the message $\mu_1^{(\ell)}$. For convenience, we complete $\mathtt{T}$, by adding to it all edges that are connected to the same variable nodes as edges that are already in $\mathtt{T}$. $\mathtt{T}^{\mathtt{c}}$ is the complement in $\{1, \cdots, n\Lambda'(1)\}$ of the set of indices $\mathtt{T}$.

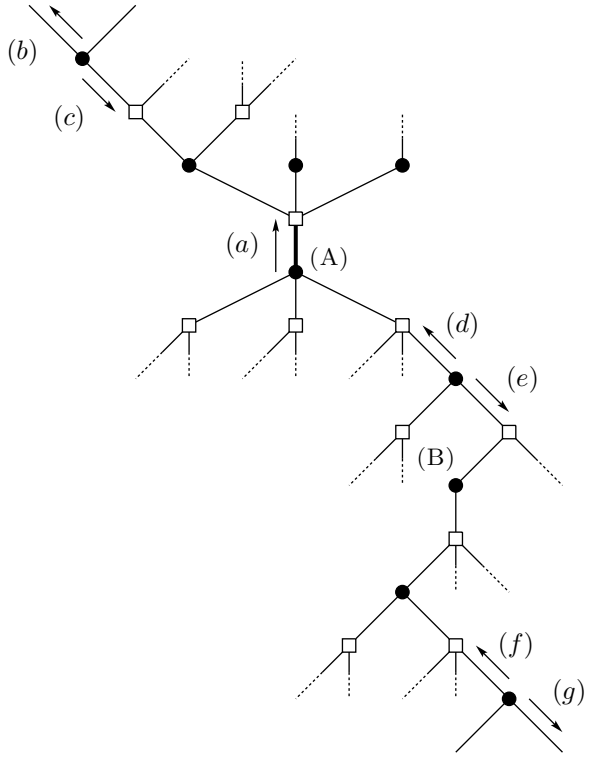


Figure 2.13: Graph representing all edges contained in $\mathtt{T}$, for the case of $\ell = 2$. The small letters represent messages sent along the edges from a variable node to a check node and the capital letters represent variable nodes. The message $\mu_1^{(\ell)}$ is represented by (a).

The set of indices $\mathtt{T}$ depends on the number of iterations performed and on the graph realization. We have depicted in Fig. 2.13 an example for the case of an irregular graph with $\ell = 2$. We have in the middle of the figure, the edge (a) carrying the message $\mu_1^{(\ell)}$. We call this edge the root edge and the variable node it is connected to, the root variable node. We will also call the message $\mu_1^{(\ell)}$ the root message. We expand the graph starting from this root node. We consider

ℓ variable node levels above the root and 2ℓ variable node levels below the root. This is due to the fact that up to ℓ levels above the root, the messages computed will depend on the value received on the root variable node, which also affects $\mu_1^{(\ell)}$ (in the figure, the value received from the channel on node (A) affects $\mu_1^{(\ell)}$ as well as the message sent on (b) after ℓ iterations). We expand 2ℓ levels in the past as the value received on a variable node at level ℓ, affects both the root edge and the edges which are 2ℓ levels below (the value received on node (B) affects both $\mu_1^{(\ell)}$ and the message sent on (g) after ℓ iterations).

We compute the two terms in (2.62) separately. Define $S = \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in \mathtt{T}} \mu_i^{(\ell)}]$ and $S^c = \lim_{n\to\infty} \left(\mathbb{E}[\mu_1^{(\ell)} \sum_{i\in \mathtt{T}^c} \mu_i^{(\ell)}] - n\Lambda'(1)x_\ell^2\right)$.

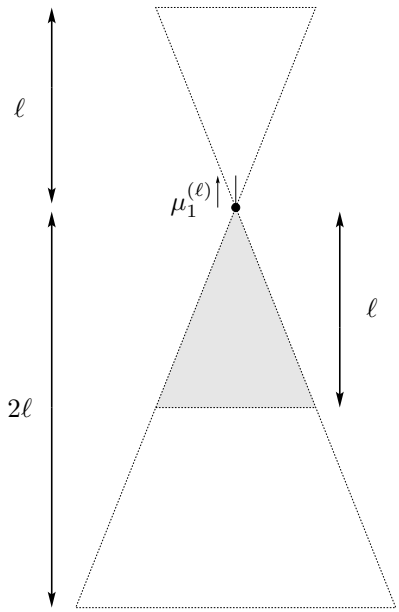


Figure 2.14: Size of $\mathtt{T}$. It contains ℓ layers of variable nodes above the root edge and 2ℓ layer of variable nodes below the root variable node. The gray area represent the computation tree of the message $\mu_1^{(\ell)}$. It contains ℓ layers of variable nodes below the root variable node.

Computation of S

Having defined $\mathtt{T}$, we can further identify four different types of terms appearing in S and write

$$\begin{aligned} S &= \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}} \mu_i^{(\ell)}] \\ &= \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)}] + \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_2} \mu_i^{(\ell)}] + \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_3} \mu_i^{(\ell)}] + \lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_4} \mu_i^{(\ell)}] \end{aligned}$$

The subset $\mathtt{T}_1$, $\mathtt{T}_1 \subset \mathtt{T}$ represents the edges above the root variable node that carry messages that point upwards (we include the root edge in $\mathtt{T}_1$). In Fig. 2.13, the message sent on edge (b) is of this type. $\mathtt{T}_2$ contains all edges that carry messages of the same type as (c), which means that they are also above the root variable node but point downwards. $\mathtt{T}_3$ represents the edges which are below the root variable node and carry messages that point upwards like edges (d) and (f). Finally, $\mathtt{T}_4$ contains all edges that are below the root variable node and point downwards like (e) and (g).

Let us start with the simplest term. In the limit of infinite blocklength, the messages carried by the edges in $\mathtt{T}_2$ at the ℓ-th iteration are independent from $\mu_1^{(\ell)}$. This gives us that

$$\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_2} \mu_i^{(\ell)}] = x_\ell^2 \rho'(1) \sum_{i=0}^{\ell-1} \rho'(1)^i \lambda'(1)^i$$

where $\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \mu_i^{(\ell)}] = x_\ell^2$ for $i \in \mathtt{T}_2$ as the two messages are independent, and the expected number of edges in $\mathtt{T}_2$ is $\rho'(1) \sum_{i=0}^{\ell-1} \lambda'(1)^i \rho'(1)^i$.

For the edges in $\mathtt{T}_1$, we write

$$\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)}] = x_\ell + x_\ell (1,0) \left(\sum_{j=1}^{\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-j+1)\mathtt{C}(\ell-j) \right) (1,0)^T, \quad (2.63)$$

with the matrices $\mathtt{V}(i)$ and $\mathtt{C}(i)$ defined in (2.58). We remind here their expressions.

$$\mathtt{V}(i) = \begin{pmatrix} \epsilon\lambda'(y_i) & 0 \\ \lambda'(1) - \epsilon\lambda'(y_i) & \lambda'(1) \end{pmatrix}, \qquad \mathtt{C}(i) = \begin{pmatrix} \rho'(1) & \rho'(1) - \rho'(\bar{x}_i) \\ 0 & \rho'(\bar{x}_i) \end{pmatrix}, \qquad i \geq 0.$$

In order to understand (2.63), consider the following case. We are at the i-th iteration of BP decoding and we pick an edge at random in the graph. It is connected to a check node of degree j with probability ρ_j. Assume further that the probability that the message carried by this edge from the variable node the check node (incoming message) is erased with probability p and known with probability $\bar{p}$. We want to compute the expected numbers of erased and known messages

sent out by the check node on its other edges (outgoing messages). If the incoming message is erased, then the number of erased outgoing messages is in expectation $(j-1)$. Averaging over the possible check node degrees gives us $\rho'(1)$. If the incoming message is known, then the expected number of erased outgoing messages is $(j-1)(1-(1-x_i)^{j-2})$. Averaging over the check node degrees gives us $\rho'(1)-\rho'(1-x_i)$. The expected number of erased outgoing messages is therefore, $p\rho'(1)+\bar{p}(\rho'(1)-\rho'(1-x_i))$. We can compute similarly, the number of known outgoing messages to find $\bar{p}\rho'(x_i)$. This can be written in a matrix form, using $\mathtt{C}(i)$. This gives us that the number of erased outgoing messages is $(1,0)\mathtt{C}(i)(p,\bar{p})^T$ and the number of known outgoing messages is $(0,1)\mathtt{C}(i)(p,\bar{p})^T$. The situation is identical if we consider a variable node instead of the check node and we use the matrix $\mathtt{V}(i)$ instead of $\mathtt{C}(i)$. We can also think of extending several layers of check and variable nodes as is shown in Fig. 2.15.

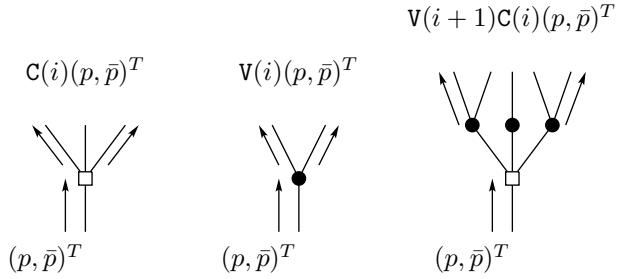


Figure 2.15: Number of outgoing erased messages as a function of the probability of erasure of the incoming message.

Now using these matrices $\mathtt{V}(i)$ and $\mathtt{C}(i)$ for $i \in \{1,\cdots,\ell\}$, we can compute the contribution of the edges in $\mathtt{T}_1$ to S. We have

$$\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)}] = \lim_{n\to\infty} \mathrm{P}\{\mu_1^{(\ell)}=1\}\mathbb{E}[\sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)} \mid \mu_1^{(\ell)}=1]. \tag{2.64}$$

The last term on the right hand side of (2.64) can be written as

$$\lim_{n\to\infty} \mathbb{E}[\sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)} \mid \mu_1^{(\ell)}=1] = 1+(1,0)\left(\sum_{j=1}^{\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-j+1)\mathtt{C}(\ell-j)\right)(1,0)^T. \tag{2.65}$$

where $\mathbb{E}[\mu_1^{(\ell)} \mid \mu_1^{(\ell)}=1]=1$. Each term of the sum, written as $(1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-j+1)\mathtt{C}(\ell-j)(1,0)^T$ for $j\in\{1,\cdots,\ell\}$, is the number of erased messages in each layer of edges in $\mathtt{T}_1$ conditioned on the fact that the root edge is erased. Now multiplying (2.65) by $\mathrm{P}\{\mu_1^{(\ell)}=1\}=x_\ell$ gives us (2.63).

The computation is similar for the edges in $\mathtt{T}_3$ and results in

$$\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_3} \mu_i^{(\ell)}] = x_\ell(1,0) \left(\sum_{j=1}^{2\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-j+1)\mathtt{C}(\ell-j) \right) (1,0)^T. \tag{2.66}$$

In this sum, when $j > \ell$, we have to evaluate the matrices $\mathtt{V}(i)$ and $\mathtt{C}(i)$ for negative indices using the definition given in (2.59).

In order to obtain S, it remains only to compute the contribution of the edges in $\mathtt{T}_4$, which is slightly more involved than computing the previous terms. $\mathtt{T}_4$ includes all the edges that are below the root node and point downwards. In Fig. 2.13, edges (e) and (g) are representative of the elements in $\mathtt{T}_4$. We claim that

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_4} \mu_i^{(\ell)}] &= (1,0) \sum_{j=0}^{\ell} \left(y_{\ell-j} U^\star(j,j) + (1-y_{\ell-j}) U^0(j,j) \right) \\ &+ (1,0) \sum_{j=\ell+1}^{2\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j) \left(y_{\ell-j} U^\star(j,\ell) + (1-y_{\ell-j}) U^0(j,\ell) \right).\end{aligned} \tag{2.67}$$

Let us consider the first term in (2.67). It corresponds to the contribution in $\mathtt{T}_4$ of edges such that the message they carry at iteration ℓ depends on the value received from the channel on the root variable node. In the case of Fig. 2.13, where $\ell = 2$, the contribution of edge (e), would be counted in this first sum. The second term in (2.67) corresponds to edges in $\mathtt{T}_4$, that are separated from the root edge by more than $\ell+1$ variable nodes. In Fig. 2.13, edge (g) is of this type.

In order to understand the first term in (2.67), consider the root edge and an edge contained in $\mathtt{T}_4$ separated from the root edge by $j+1$ variable node with $j \in \{0,\cdots,\ell\}$. For this edge in $\mathtt{T}_4$, we consider two messages it carries. The message that is sent from the variable node to the check node at the ℓ-th iteration (outgoing) and that participate in our second moment calculation and the message sent from the check node to the variable node at the $(\ell-j)$-th iteration (incoming). We define the two-components vector $U^\star(j,j)$, such that the first component is the joint probability that both the root and the outgoing messages are erased conditioned on the fact that the incoming message is erased, multiplied by the number of edges in $\mathtt{T}_4$ which are similar to the edge of interest (at the same distance from the root edge). The second component is the joint probability that the root message is erased and that the outgoing message is known conditioned on the fact that the incoming message is erased, again multiplied by the number of edges in $\mathtt{T}_4$ which are similar to the edge of interest. The vector $U^0(j,j)$ is defined in exactly the same manner except that the incoming message on which the joint probability is conditioned is

in this case known. Therefore, the superscript $\star$ or 0 accounts respectively for the cases where the incoming message is erased or known. From these definitions, it is clear that the contribution to S of the edges that are in $\mathtt{T}_4$ and separated from the root edge by $j+1$ variable nodes with $j \in \{0, \cdots, \ell\}$, is written as $(1,0)\left(y_{\ell-j}U^\star(j,j) + (1-y_{\ell-j})U^0(j,j)\right)$. We still have to evaluate $U^\star(j,j)$ and $U^0(j,j)$ for any $j \in \{0, \cdots, \ell\}$. In order to do this, we have to define the vectors $U^\star(j,k)$ and $U^0(j,k)$ with $k \leq j$, which are similar to the previous quantities, except that this time we look at the root edge and an edge in $\mathtt{T}_4$ separated from the root edge by $k+1$ variable nodes. The outgoing message we consider is the one at the $(\ell - j + k)$-th iteration and the incoming message we condition on, is the one at the $(\ell - k)$-th iteration. It is easy to check that $U^\star(j,j)$ and $U^0(j,j)$ can be computed in a recursive manner using $U^\star(j,k)$ and $U^0(j,k)$. The initial conditions are

$$\begin{aligned} U^\star(j,0) =& (y_{\ell-j}\epsilon\lambda'(y_\ell), (1-y_{\ell-j})\epsilon\lambda'(y_\ell))^T, \\ U^0(j,0) =& (0,0)^T, \end{aligned}$$

and the recursion is for $k \in \{1, \cdots, j\}$ is the one given in Lemma 9. In this case, the computation trees of both messages we consider, the root message and the message carried by the edge in $\mathtt{T}_4$ overlap and as the two edges are separated by at most $\ell+1$ variable nodes, any received value which is on a path between the two edges affects both their values. This is why this recursion is slightly more involved than the one to compute the contribution of the edges in $\mathtt{T}_1$. The situation is depicted in the left side of Fig. 2.16.

For the case of edges in $\mathtt{T}_4$ that are separated from the root edge by more than $\ell+1$ variable nodes. The situation is slightly different and is depicted in the right side of Fig. 2.16. In this case, if we consider any path between the two edges. Some of the received values will affect both messages (the ones which are in the intersection of the computation trees) and others will affect only one of the messages. We therefore have to adapt the previous recursion. We start from the root edge and compute the effect of the received values that only affect this message resulting in a expression similar to the one we used to compute the contribution of $\mathtt{T}_1$. This gives us the following initial conditions.

$$\begin{aligned} U^\star(j,j-\ell) =& (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)(1,0)^T M_1(j,j-\ell)(y_{\ell-j}, 1-y_{\ell-j})^T \\ &+ (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)(0,1)^T M_2(j,j-\ell)(y_{\ell-j}, 1-y_{\ell-j})^T, \\ U^0(j,j-\ell) =& (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)(0,1)^T\mathtt{V}(0)(y_{\ell-j}, 1-y_{\ell-j})^T. \end{aligned}$$

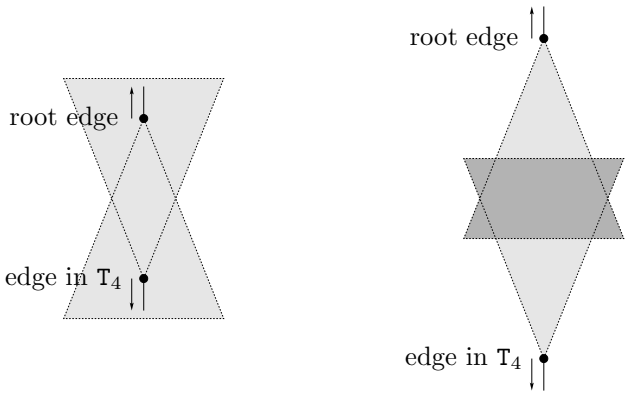


Figure 2.16: The two situations that arise when computing the contribution of $\mathtt{T}_4$. In the left side we show the case where the two edges are separated by at most $\ell+1$ variable nodes and in the right side, the case where they are separated by more than $\ell+1$ variable nodes.

We apply the recursion given in Lemma 9 to the intersection of the computation trees. We have to stop the recursion at $k=\ell$ (end of the intersection of the computation trees). It remains to account for the received values that only affect the messages on the edge in $\mathtt{T}_4$. This is done by writing

$$(1,0)\sum_{j=\ell+1}^{2\ell}\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(2\ell-j+1)\mathtt{C}(2\ell-j)\left(y_{\ell-j}U^{\star}(j,\ell)+(1-y_{\ell-j})U^{0}(j,\ell)\right),$$

which is the second term of (2.67).

Computation of S^c

We still need to compute $S^c=\lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}^c}\mu_i^{(\ell)}]-n\Lambda'(1)x_\ell^2\right)$. Recall that by definition, all the messages that are carried by edges in $\mathtt{T}^c$ at the ℓ-th iteration are computed based on received values that do not participate in the computation of $\mu_1^{(\ell)}$. At first sight, one might think that these messages are independent of $\mu_1^{(\ell)}$. This is indeed the case when the bipartite graph is regular (all of the variable nodes have the same degree as do all of the check nodes).

For the degree distributions $\lambda(x) = x^{\mathtt{l}-1}$ and $\rho(x) = x^{\mathtt{r}-1}$, we have that

$$\begin{aligned} S^c &= \lim_{n\to\infty} \left(\mathbb{E}[\mu_1^{(\ell)} \sum_{i\in \mathtt{T}^c} \mu_i^{(\ell)}] - n\Lambda'(1)x_\ell^2 \right) \\ &= \lim_{n\to\infty} \left(|\mathtt{T}^c| x_\ell^2 - n\Lambda'(1)x_\ell^2 \right) \\ &= \lim_{n\to\infty} \left((n\Lambda'(1) - |\mathtt{T}|) x_\ell^2 - n\Lambda'(1)x_\ell^2 \right) \\ &= -|\mathtt{T}| x_\ell^2 \end{aligned}$$

with the cardinality of $\mathtt{T}$ being $|\mathtt{T}| = \sum_{i=0}^{2\ell} (\mathtt{l}-1)^i(\mathtt{r}-1)^i\mathtt{l} + (\mathtt{r}-1)\sum_{i=1}^{\ell} (\mathtt{l}-1)^{i-1}(\mathtt{r}-1)^{i-1}\mathtt{l}$. However, when we consider irregular ensembles, the cardinality of $\mathtt{T}$ is not fixed anymore and depends on the graph realization. If we call $\mathtt{G}_\mathtt{T}$ the graph composed of the edges in $\mathtt{T}$ and of the variable and check nodes connecting them, it is clear that the message $\mu_1^{(\ell)}$ depends on the realization of $\mathtt{G}_\mathtt{T}$. We will also see that the messages carried by the edges in $\mathtt{T}^c$ also depend on the realization of $\mathtt{G}_\mathtt{T}$. We write

$$\begin{aligned} S^c &= \lim_{n\to\infty} \left(\mathbb{E}[\mu_1^{(\ell)} \sum_{i\in \mathtt{T}^c} \mu_i^{(\ell)}] - n\Lambda'(1)x_\ell^2 \right), \\ &= \lim_{n\to\infty} \left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[\mathbb{E}[\mu_1^{(\ell)} \sum_{i\in \mathtt{T}^c} \mu_i^{(\ell)} \mid \mathtt{G}_\mathtt{T}]] - n\Lambda'(1)x_\ell^2 \right), \\ &= \lim_{n\to\infty} \left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[\mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G}_\mathtt{T}] |\mathtt{T}^c| \mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]] - n\Lambda'(1)x_\ell^2 \right), && j \in \mathtt{T}^c, \\ &= \lim_{n\to\infty} \left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[\mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G}_\mathtt{T}] (n\Lambda'(1) - |\mathtt{T}|) \mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]] - n\Lambda'(1)x_\ell^2 \right), && j \in \mathtt{T}^c, \\ &= \lim_{n\to\infty} n\Lambda'(1) \left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[\mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G}_\mathtt{T}] \mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]] - n\Lambda'(1)x_\ell^2 \right), && j \in \mathtt{T}^c, \qquad (2.68) \\ &\quad - \lim_{n\to\infty} \mathbb{E}_{\mathtt{G}_\mathtt{T}}[|\mathtt{T}| \mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G}_\mathtt{T}] \mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]], && j \in \mathtt{T}^c. \qquad (2.69) \end{aligned}$$

We need therefore to compute $\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]$ for a fixed realization of $\mathtt{G}_\mathtt{T}$ and a random edge j taken from $\mathtt{T}^c$. This value differs slightly from x_ℓ due to two effects. The first is related to the fact that we are dealing with a fixed-size bipartite graph (although we later take the limit $n \to \infty$) and therefore the degrees of the nodes that are in $\mathtt{G}_\mathtt{T}$ affect the degrees of the nodes that are in its complement (with respect to the total bipartite graph). Intuitively, if $\mathtt{G}_\mathtt{T}$ contains an unusually large number of high degree variable nodes, the rest of the graph will correspondingly contain an unusually small number of high degree variable nodes affecting the average $\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]$. This first effect can be measured by computing the degree distribution over the rest of the graph as a function of the characteristics of $\mathtt{G}_\mathtt{T}$. The second effect is due to the fact that the messages that flow out of $\mathtt{G}_\mathtt{T}$ affect the message that are computed on edges in $\mathtt{T}^c$ which are close to $\mathtt{G}_\mathtt{T}$. We will

measure this effect by looking in detail to the messages that flow out of $\mathtt{G_T}$. Let us consider for the moment the first effect. Define $V^{\mathtt{G_T}}(x) = \sum_i V_i^{\mathtt{G_T}} x^i$ and $C^{\mathtt{G_T}}(x) = \sum_j C_j^{\mathtt{G_T}} x^j$ such that $V_i^{\mathtt{G_T}}$ is the number of variable nodes of degree i and $C_j^{\mathtt{G_T}}$ is the number of check nodes of degree j in $\mathtt{G_T}$. The derivatives of these functions are $V^{\mathtt{G_T}'}(x) = \sum_i i V_i^{\mathtt{G_T}} x^{i-1}$ and $C^{\mathtt{G_T}'}(x) = \sum_j j C_j^{\mathtt{G_T}} x^{j-1}$ respectively. It is easy to verify that if we take a bipartite graph having a variable degree distribution $\lambda(x)$ and remove a variable node of degree i, the variable degree distribution changes by

$$\Delta_i \lambda(x) = \frac{i\lambda(x) - i x^{i-1}}{n\Lambda'(1)} + O(1/n^2).$$

Therefore, if we remove $\mathtt{G_T}$ from the bipartite graph, the remaining graph will have a variable perspective degree distribution that differs from the original by

$$\Delta\lambda(x) = \frac{V^{\mathtt{G_T}'}(1)\lambda(x) - V^{\mathtt{G_T}'}(x)}{n\Lambda'(1)} + O(1/n^2).$$

In the same way, removing $\mathtt{G_T}$ changes the check degree distribution by

$$\Delta\rho(x) = \frac{C^{\mathtt{G_T}'}(1)\rho(x) - C^{\mathtt{G_T}'}(x)}{n\Lambda'(1)} + O(1/n^2).$$

The resulting change in the fraction of erased messages sent from the variable to the check nodes can be easily calculated. If the degree distributions change slightly by $\Delta\lambda(x)$ and $\Delta\rho(x)$, x_ℓ changes by Δx_ℓ such that

$$\begin{aligned}
\Delta x_\ell =& \sum_{i=1}^{\ell} \prod_{k=i+1}^{\ell} \epsilon\lambda'(y_k)\rho'(1-x_{k-1}) \left(\epsilon\Delta\lambda(y_i) - \epsilon\lambda'(y_i)\Delta\rho(1-x_{i-1})\right), \\
=& \frac{1}{n\Lambda'(1)} \sum_{i=1}^{\ell} \prod_{k=i+1}^{\ell} \epsilon\lambda'(y_k)\rho'(1-x_{k-1}) \Big(\epsilon(V^{\mathtt{G_T}'}(1)\lambda(y_i) - V^{\mathtt{G_T}'}(y_i)) \\
& -\epsilon\lambda'(y_i)(C^{\mathtt{G_T}'}(1)\rho(1-x_{i-1}) - C^{\mathtt{G_T}'}(1-x_{i-1}))\Big) + O(1/n^2), \\
=& \frac{1}{n\Lambda'(1)} \sum_{i=1}^{\ell} F_i \left(x_i V^{\mathtt{G_T}'}(1) - \epsilon V^{\mathtt{G_T}'}(y_i) - \epsilon\lambda'(y_i)(C^{\mathtt{G_T}'}(1)\rho(1-x_{i-1}) - C^{\mathtt{G_T}'}(1-x_{i-1}))\right) + O(1/n^2),
\end{aligned}$$

with $F_i = \prod_{k=i+1}^{\ell} \epsilon\lambda'(y_k)\rho'(1-x_{k-1})$ as defined in (2.60).

Consider now the second effect of a particular choice of $\mathtt{G_T}$ over $\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G_T}]$. As already stated, this expectation is also affected by the messages that flow out of the boundary of $\mathtt{G_T}$. Call $\mathcal{B}$ the number of edges forming this boundary (edges emanating upwards from the variable nodes that are ℓ levels above the root edge and emanating downwards from the variable nodes that are 2ℓ levels below the root variable node) and call $\mathcal{B}_i^\star$ the number of erased messages carried at the i-th iteration by these edges. As a result, if $\tilde{x}_i$ is the fraction of erased messages incoming to the

check nodes in the complement of $\mathtt{G_T}$ at the i-th iteration, taking into account this effect gives a new erasure fraction $\hat{x}_i$ that can be written as

$$\begin{aligned}\hat{x}_i =& \frac{(n\Lambda'(1)-\mathcal{B})\tilde{x}_i + \mathcal{B}_i^\star}{n\Lambda'(1)} + O(1/n^2),\\ =& \tilde{x}_i + \frac{\mathcal{B}_i^\star - \mathcal{B}\tilde{x}_i}{n\Lambda'(1)} + O(1/n^2).\end{aligned}$$

This expression simply comes from the fact that at the i-th iteration, we have $(n\Lambda'(1)-\mathcal{B})$ message in the complement of $\mathtt{G_T}$ of which a fraction $\tilde{x}_i$ is erased. In the $\mathcal{B}$ remaining messages, there are $\mathcal{B}_i^\star$ erasures, which gives us the expression. Consequently, combining the above two effects, we can write that at the ℓ-th iteration, the fraction of erased messages that are sent by the variable nodes outside $\mathtt{G_T}$ can be written, for $j \in \mathtt{T^c}$, as

$$\begin{aligned}\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G_T}] =& x_\ell + \frac{1}{n\Lambda'(1)}\sum_{i=1}^{\ell} F_i\left(x_i V^{\mathtt{G_T}'}(1) - \epsilon V^{\mathtt{G_T}'}(y_i) - \epsilon\lambda'(y_i)(C^{\mathtt{G_T}'}(1)\rho(1-x_{i-1}) - C^{\mathtt{G_T}'}(1-x_{i-1}))\right)\\ &+ \frac{1}{n\Lambda'(1)}\sum_{i=1}^{\ell-1} F_i\left(\mathcal{B}_i^\star - \mathcal{B}x_i\right) + O(1/n^2).\end{aligned}$$

We can now use this expression in (2.68) and (2.69) to obtain

$$\begin{aligned}S^c =& \lim_{n\to\infty} n\Lambda'(1)\left(\mathbb{E}_{\mathtt{G_T}}[\mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G_T}]\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G_T}]] - n\Lambda'(1)x_\ell^2\right), && j \in \mathtt{T^c},\\ &- \lim_{n\to\infty} \mathbb{E}_{\mathtt{G_T}}[|\mathtt{T}|\mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G_T}]\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G_T}]], && j \in \mathtt{T^c},\\ =& \sum_{i=1}^{\ell} F_i\left(x_i\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G_T}'}(1)] - \epsilon\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G_T}'}(y_i)]\right)\\ &- \sum_{i=1}^{\ell} F_i\epsilon\lambda'(y_i)\left(\mathbb{E}[\mu_1^{(\ell)}C^{\mathtt{G_T}'}(1)]\rho(1-x_{i-1}) - \mathbb{E}[\mu_1^{(\ell)}C^{\mathtt{G_T}'}(1-x_{i-1})]\right)\\ &+ \sum_{i=1}^{\ell-1} F_i\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star]\\ &- \sum_{i=1}^{\ell-1} F_i x_i\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}]\\ &- x_\ell\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G_T}'}(1)],\end{aligned}$$

where we took the limit $n\to\infty$, removed the terms tending to zero and finally, we replaced $|\mathtt{T}|$ by $V^{\mathtt{G_T}'}(1)$.

It is clear what each of these values represent. For example, $\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G_T}'}(1)]$ is the expected value of $\mu_1^{(\ell)}$ multiplied by the number of edges that are in $\mathtt{G_T}$. Each of these terms can be

computed through recursions that are similar in spirit to the ones used to compute S. These recursions are provided in the body of Lemma 9. We will just explain in further detail how the terms $\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}]$ and $\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star]$ can be computed. We claim that the first term can be written as

$$\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}] = \left(x_\ell + (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(1)\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)\left(\lambda'(1)\rho'(1)\right)^\ell.$$

This is because the value of $\mu_1^{(\ell)}$ depends only on the realization of its computation tree and not on the realization of the whole $\mathtt{G}_\mathtt{T}$. From the definitions of $\mathtt{G}_\mathtt{T}$ and the computation tree, we have that the boundary of $\mathtt{G}_\mathtt{T}$ is on average $(\lambda'(1)\rho'(1))^\ell$ larger than the boundary of the computation tree of $\mu_1^{(\ell)}$. Finally, the expectation of $\mu_1^{(\ell)}$ multiplied by the number of edges in the boundary of its computation tree can be easily computed as $\left(x_\ell + (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(1)\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)$. Multiplying these two terms give us our expression. For $\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star]$, the situation is similar. We can start by computing the expectation of $\mu_1^{(\ell)}$ multiplied by the number of edges in the boundary of its computation tree. This number has to be multiplied by $(1,0)\mathtt{V}(i)\mathtt{C}(i-1)\cdots\mathtt{V}(i-\ell+1)\mathtt{C}(i-\ell)(1,0)^T$ to account for what happens between the boundary of the computation tree and the boundary of $\mathtt{G}_\mathtt{T}$. We hence obtain

$$\begin{aligned}\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star] = &\left(x_\ell + (1,0)\mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(1)\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)\\ &\cdot(1,0)\mathtt{V}(i)\mathtt{C}(i-1)\cdots\mathtt{V}(i-\ell+1)\mathtt{C}(i-\ell)(1,0)^T.\end{aligned}$$

□

The expression provided in the above lemma has been used to plot $\mathcal{V}^{(\ell)}$ for $\epsilon \in (0,1)$ and for several values of ℓ in the case of an irregular ensemble in Fig. 2.4.

It remains to determine the asymptotic behavior of this quantity as the number of iterations converges to infinity.

Lemma 10. Let $\mathtt{G}$ be chosen uniformly at random from LDPC(n,λ,ρ) and consider transmission over the BEC of erasure probability ϵ. Label the $n\Lambda'(1)$ edges of $\mathtt{G}$ in some fixed order by the elements of $\{1,\cdots,n\Lambda'(1)\}$. Assume that we decode until no further progress can be accomplished. Set $\mu_i^{(\infty)}$ equal to one if the message along edge i from the variable to the check

node side is an erasure and equal to zero otherwise. Then

$$\lim_{\ell\to\infty}\lim_{n\to\infty}\frac{\mathbb{E}[(\sum_i \mu_i^{(\infty)})^2]-\mathbb{E}[(\sum_i \mu_i^{(\infty)})]^2}{n\Lambda'(1)} =$$
$$+\frac{\epsilon^2\lambda'(y)^2(\rho(\bar{x})^2-\rho(\bar{x}^2)+\rho'(\bar{x})(1-2x\rho(\bar{x}))-\bar{x}^2\rho'(\bar{x}^2))}{(1-\epsilon\lambda'(y)\rho'(\bar{x}))^2}$$
$$+\frac{\epsilon^2\lambda'(y)^2\rho'(\bar{x})^2(\epsilon^2\lambda(y)^2-\epsilon^2\lambda(y^2)-y^2\epsilon^2\lambda'(y^2))}{(1-\epsilon\lambda'(y)\rho'(\bar{x}))^2}$$
$$+\frac{(x-\epsilon^2\lambda(y^2)-y^2\epsilon^2\lambda'(y^2))(1+\epsilon\lambda'(y)\rho'(\bar{x}))+\epsilon y^2\lambda'(y)}{1-\epsilon\lambda'(y)\rho'(\bar{x})}.$$

E - Proof of Lemma 5

In this Appendix we present a proof of Lemma 5, making use of Doob's maximal inequality (2.27). We shall prove that each of the two events considered in Eq. (2.28) occurs with probability greater than $1-\Omega_1\exp[-\Omega_2\,\delta^2]$. This implies the thesis by a simple union bound, plus a rescaling of the constants Ω_1, Ω_2.

Let us begin by considering the second event, namely $X^{(0)}_{t_g}\geq X^{(0)}_{t^*}-\delta^{4/3}\,n^{1/3}$. For sake of simplicity we redefine $t_{\rm g}$ to be the position of the global minimum of $X^{(0)}_t$ in the domain $t>t^*$. The minimum with an unrestricted t can be treated by putting together the cases $t>t^*$ and $t<t_*$. It is also useful to define

$$Y_{t-t_*}:=\frac{1}{\kappa_1}(X^{(0)}_t-X^{(0)}_{t^*})\,.$$

Equation (2.27) implies

$$\mathrm{P}\left\{\min_{0\leq t\leq T}\left[Y_t-\frac{1}{n}\,t^2+\frac{\kappa_2\delta}{\sqrt{n}}\,t\right]\leq-\delta\sqrt{T}\right\}\leq\Omega_1\,e^{-\Omega_2\delta^2}\,,\tag{2.70}$$

where we rescaled the constants κ_2 and Ω_2.

Let $\{t_l:l\in\mathbb{Z}\}$ be a non-decreasing sequence of real numbers with $t_l\to\infty$ as $l\to\infty$ and $t_l=0$ as $l\to-\infty$. A union bound yields

$$\mathrm{P}\left\{\min_{t\geq0}\,Y_t\leq-\delta^{4/3}n^{1/3}\right\}\leq\sum_{l=-\infty}^{+\infty}\mathrm{P}\left\{\min_{t_l\leq t<t_{l+1}}Y_t\leq-\delta^{4/3}n^{1/3}\right\}\leq$$
$$\leq\sum_{l=-\infty}^{+\infty}\mathrm{P}\left\{\min_{t_l\leq t<t_{l+1}}\left[Y_t-\frac{1}{n}\,t^2+\frac{\kappa_2\delta}{\sqrt{n}}\,t\right]\leq-\delta^{4/3}n^{1/3}-\frac{1}{n}\,t_l^2+\frac{\kappa_2\delta}{\sqrt{n}}\,t_{l+1}\right\}\leq$$
$$\leq\Omega_1\sum_{l=-\infty}^{+\infty}\exp\left\{-\Omega_2\frac{1}{t_{l+1}}\left(\delta^{4/3}n^{1/3}+\frac{1}{n}\,t_l^2-\frac{\kappa_2\delta}{\sqrt{n}}\,t_{l+1}\right)\right\},$$

where we used Eq. (2.70) in the last inequality. At thin point we choose $t_l = 2^l(n\delta)^{2/3}$. Plugging into the above expression we get

$$\mathrm{P}\left\{\min_{t\geq 0} Y_t \leq -\delta^{4/3}n^{1/3}\right\} \leq \Omega_1 \sum_{l=-\infty}^{+\infty} \exp\left\{-\frac{\Omega_2\delta^2}{2^{l+1}}\left(1+2^{2l}-\frac{\kappa_2\delta^{1/3}}{n^{1/6}}2^{l+1}\right)^2\right\}.$$

If $n > n_0(\delta) := (2\kappa_2)^6\delta^2$ we get

$$\mathrm{P}\left\{\min_{t\geq 0} Y_t \leq -\delta^{4/3}n^{1/3}\right\} \leq \Omega_1 \sum_{l=-\infty}^{+\infty} \exp\left\{-\frac{\Omega_2\delta^2}{2^{l+1}}\left(1+2^{2l}-2^l\right)^2\right\}.$$

It is an elementary exercise to show that the right hand side is smaller than $\Omega_1' \exp\{-\Omega_2'\delta^2\}$ for some (eventually different) positive parameters Ω_1' and Ω_2' and any $\delta > \delta_0$.

The second part of the proof consists in proving an analogous upper bound for the probability of having $|t_{\rm g} - t^*| > \delta^{2/3}n^{2/3}$. In fact the proof proceeds as for the first event. One splits the semi-infinite interval $t > t^*$ in intervals $[t_l, t_{l+1}[$ with $t_l = 2^l(n\delta)^{2/3}$ and (this time) $l \geq 0$, and then apply Doob's maximal inequality to each interval. We leave to the reader the pleasure of filling the details.

F - Convergence to diffusion process

In this Appendix we prove Lemma 6 as a straightforward application of the following statement which can be found in [26].

Theorem 1. *Let $\{X_t\}$ be a Markov process with values in $\mathbb{R}^d$ and transition probability $\pi_h(x, dy)$, with $0 < h \leq 1$ and initial condition $X_0 = x_0$. Let P_h be the measure induced on the space of continuous trajectories $\Omega = C([0,\infty), \mathbb{R}^d)$ by the mapping $X(th) = X_t$ for integer t and interpolating linearly in between. Assume that the limit*

$$\lim_{h\to\infty} \frac{1}{h}\int_{\mathbb{R}^d} [\phi(y) - \phi(x)]\, \pi_h(x, dy) = (\mathcal{L}\phi)(x)\,, \tag{2.71}$$

exists uniformly in a compact $K \subseteq \mathbb{R}^d$ for functions $\phi \in C^\infty(K)$. Assume that the limit has the form

$$(\mathcal{L}\phi)(x) = \frac{1}{2}\sum_{ij} a_{ij}(x)\frac{\partial^2\phi}{\partial x_i \partial x_j} + \sum_{i=1}^{d} b_i(x)\frac{\partial\phi}{\partial x_i}\,, \tag{2.72}$$

with continuous and uniformly bounded coefficients $a \equiv \{a_{ij}(x)\}$ (a being a positive definite matrix) and $b \equiv \{b_i(x)\}$. Assume finally that the solution of the martingale problem for $\mathcal{A}$ is unique yielding a Markov family of measures P_x on Ω. Then $\{P_h, x\}$ converges to $\{P_x\}$ as $h \to 0$.

The proof of Lemma 6 proceed then as follows. Set $h = n^{-2/3}$ and define the a Markov chain in the variables $u_0, \vec{u}$, see Eq. (2.31), (2.31) using the transition rates $W(\Delta|x)$ and the initial condition $u_0(0) = \zeta$, $\vec{u}(0) = 0$. One has then just to compute the generator

$$(\mathcal{L}\phi)(u_0, \vec{u}) = \lim_{n\to\infty} n^{2/3} \sum_{\Delta_0, \vec{\Delta}} [\phi(u_0 + n^{-1/3}\Delta_0, \vec{u} + n^{-2/3}\vec{\Delta}) - f(u_0, \vec{u})] \cdot \widehat{W}(\Delta_0, \vec{\Delta}|n^{-2/3}v_0, n^{-1}\vec{X}_{t_*} + n^{-1/3}\vec{u}), \tag{2.73}$$

where made the substitution $W(\Delta|x) \to \widehat{W}(\Delta|x/n)$ which implies a negligible $O(1/n)$ error. The formula (2.33) is easily obtained by Taylor expansion the above equation.

Chapter 3

Error Floor

3.1 Motivation

The error probability curves of LDPC codes can be split in two part. In the previous chapter, we considered large error events (linear-sized) resulting in the waterfall region. In this chapter, we consider the error floor region due to small errors remaining after the decoding.

In the following, we will show that knowing the expected number of stopping sets of each size present in a randomly chosen graph from a given ensemble is sufficient to derive a good approximation of the error floor. As we will see, the key to a good approximation is to look at *minimal* stopping sets (stopping sets that are not the union of smaller stopping sets) and their distribution as the blocklength increases. The resulting approximation is shown in Fig. 3.1 for a specific example.

3.2 Expected Number of Stopping Sets

Consider elements from the ensemble LDPC$(n, \lambda(x), \rho(x))$ and transmission over a BEC of erasure probability ϵ. Call $A_{\mathtt{G}}(s)$ the number of stopping sets of size s in the bipartite graph $\mathtt{G}$. It has been shown in [16], that the expected value of $A_{\mathtt{G}}(s)$ can be computed easily. We call this value $A_s = \mathbb{E}[A_{\mathtt{G}}(s)]$. Let us start by recalling how this computation is done. Consider s variable nodes in the graph having e edges emanating from them. The number of such sets of variable

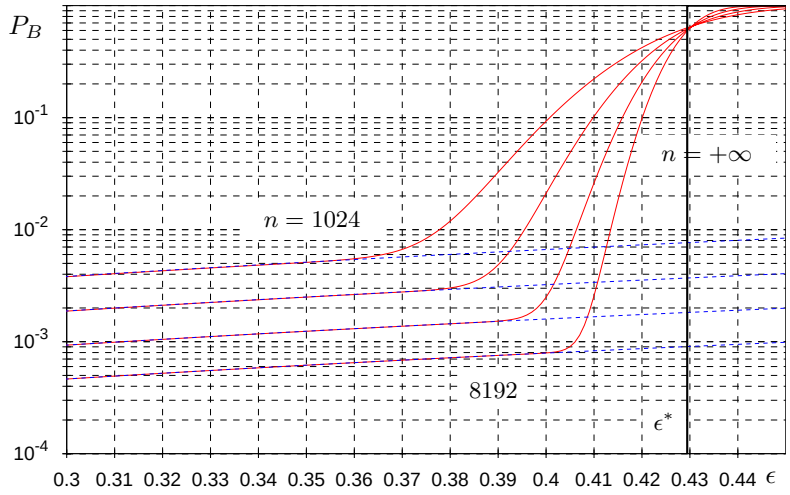


Figure 3.1: Block error probability curves for LDPC$(n, \lambda(x) = x^2, \rho(x) = x^5)$ when used over a binary erasure channel of erasure fraction ϵ. The different curves are for $n \in \{1024, 2048, 4096, 8192\}$. The threshold is $\epsilon^* \simeq 0.42943981$. The solid curves are the exact block error probabilities and the dashed curves are obtained through our approximation given in (3.8).

nodes in any graph is equal to

$$\text{coef}\left\{\prod_i (1+xy^i)^{n\Lambda_i}, x^s y^e\right\}. \tag{3.1}$$

Let us now consider one such set of variable nodes. Its edges can be connected to the check nodes in $\binom{n\Lambda'(1)}{e}$ different ways. Among these, let us count the number of constructions giving rise to stopping sets. We know that if a check node is connected to a stopping set, then it should be connected to it at least twice. This gives us that the number of stopping sets that one can construct with these e edges is

$$\text{coef}\left\{\prod_i ((1+x)^i - ix)^{n_c P_i}, x^e\right\}.$$

Therefore, the probability that a set of variable nodes of size s having e emanating edges is a stopping set is

$$\frac{\text{coef}\left\{\prod_i ((1+x)^i - ix)^{n_c P_i}, x^e\right\}}{\binom{n\Lambda'(1)}{e}}.$$

In order to obtain the expected number of stopping sets of size s having e edges in a graph, it remains to multiply this probability with the number of such sets (expression (3.1)). Finally, we

have to sum over all e as we are are interested in the expected number of stopping sets of size s regardless of the number of edges involved. This gives us

$$A_s = \sum_e \text{coef}\left\{\prod_i (1+xy^i)^{n\Lambda_i}, x^s y^e\right\} \frac{\text{coef}\left\{\prod_i ((1+x)^i - ix)^{n_c P_i}, x^e\right\}}{\binom{n\Lambda'(1)}{e}}. \tag{3.2}$$

We show in the Appendix that computing these values for a fixed size s can be done with a complexity that is independent of the blocklength.

It is tempting to use a simple union bound to approximate the bit error probability resulting from small stopping sets. The bound would read

$$\frac{1}{n}\sum_{s\geq 1} sA_s\epsilon^s,$$

where ϵ^s is the probability that a stopping set is erased and s is the number of errors we make in that case. Several problems arise. It is clear that this union bound overestimates the probability of error. For example, in A_2, we count the average number of stopping sets of size 2 including those that are the union of two stopping sets of size 1. But those stopping sets of size 1 have already been counted through A_1 and therefore are overcounted in our sum.

Further, if we want to bound the probability that there is no stopping sets of a fixed size S in the set of erased variables nodes, it is not enough just to know the average values A_s but we need to know the distribution of number of stopping sets.

Fortunately, knowing the distribution of minimal stopping sets (stopping sets that are not union of smaller stopping sets) in a randomly chosen graph will help us to address both issues. The asymptotic distribution of minimal stopping sets of LDPC codes was studied in [20].

3.3 Asymptotic Distribution of Minimal Stopping Sets

Let us start by recalling the results in [20] about the asymptotic distribution of minimal stopping sets for LDPC codes. The analysis leading to these results is reminiscent of the study of the distribution of cycles in random graphs (see [30]).

Consider elements from the ensemble LDPC$(n, \lambda(x), \rho(x))$, whose degree distribution has a non-zero fraction of degree 2 variable nodes. Define $\mu \triangleq \lambda'(0)\rho'(1)$ and $\mu_i = \frac{\mu^i}{2i}$. Let $\tilde{A}_{\mathcal{G}}(s)$, $s \in \mathbb{N}$, be the number of minimal stopping sets of size s present in a random element of the

ensemble. Then for $S \in \mathbb{N}$ we have

$$\lim_{n\to+\infty} \mathrm{P}\left\{\tilde{A}_{\mathtt{G}}(1) = \tilde{a}_1, \cdots, \tilde{A}_{\mathtt{G}}(S) = \tilde{a}_S\right\} = \prod_{s=1}^{S} \frac{\mu_s^{\tilde{a}_s} e^{-\mu_s}}{\tilde{a}_s!}.$$

In words, this means that the distribution of stopping sets of sizes between 1 and S in a random element of the ensemble tends to a joint Poisson distribution with independent components and means $(\mu_1, \cdots, \mu_S)$, for any fixed integer S. Building on this result, the bit error probability due to stopping sets of sizes between $s_{\min}$ and S call it $\tilde{\mathrm{P}}^{E}_{\mathrm{b},s_{\min},S}(n, \lambda(x), \rho(x), \epsilon)$ and the corresponding block error probability $\tilde{\mathrm{P}}^{E}_{\mathrm{B},s_{\min},S}(n, \lambda(x), \rho(x), \epsilon)$ are asymptotically given by

$$\lim_{n\to+\infty} n\tilde{\mathrm{P}}^{E}_{\mathrm{b},s_{\min},S}(n, \lambda(x), \rho(x), \epsilon) = \sum_{s=s_{\min}}^{S} s\mu_s\epsilon^s = \frac{1}{2}\sum_{s=s_{\min}}^{S} (\mu\epsilon)^s, \tag{3.3}$$

$$\lim_{n\to+\infty} \tilde{\mathrm{P}}^{E}_{\mathrm{B},s_{\min},S}(n, \lambda(x), \rho(x), \epsilon) = 1 - e^{-\sum_{s=s_{\min}}^{S} \frac{(\mu\epsilon)^s}{2s}}. \tag{3.4}$$

Equation (3.3) corresponds to simply counting the average number of minimal stopping sets of all sizes between $s_{\min}$ and S weighted by their size. Equation (3.6) is only slightly more complex. The probability that there is a block error is the complement of the probability that all minimal stopping sets present in the graph are not erased by the channel. Using the asymptotic distribution we can write that the probability that there is $\tilde{a}_s$ stopping set in the graph and that none of them is erased is $\frac{\mu_s^{\tilde{a}_s} e^{-\mu_s}}{\tilde{a}_s!}\left(1-\epsilon^s\right)^{\tilde{a}_s}$. Taking the product for all s and summing over all possible $\tilde{a}_s$ gives us the result.

$$\begin{aligned}
&1 - \sum_{\tilde{a}_1,\cdots,\tilde{a}_S} \prod_{s=s_{\min}}^{S} \frac{\mu_s^{\tilde{a}_s} e^{-\mu_s}}{\tilde{a}_s!}\left(1-\epsilon^s\right)^{\tilde{a}_s} \\
=&1 - \prod_{s=s_{\min}}^{S} e^{-\mu_s} \sum_{\tilde{a}_s} \frac{\left(\mu_s(1-\epsilon^s)\right)^{\tilde{a}_s}}{\tilde{a}_s!} \\
=&1 - \prod_{s=s_{\min}}^{S} e^{-(\mu_s\epsilon^s)} \\
=&1 - e^{-\sum_{s=s_{\min}}^{S} \frac{(\mu\epsilon)^s}{2s}}
\end{aligned}$$

For the case where $\mu\epsilon < 1$, it is also shown in [20] that the error probability due to all stopping sets that are between $s_{\min}$ and $n\gamma\nu^*$, such that $\gamma \in (0,1)$ and ν^* is the fractional size of the residual graph at the threshold, call it $\tilde{\mathrm{P}}^{E}_{\mathrm{b},s_{\min},\gamma}(n, \lambda(x), \rho(x), \epsilon)$ for the bit error probability and

$\tilde{\mathrm{P}}^{E}_{\mathrm{B},s_{\min},\gamma}(n,\lambda(x),\rho(x),\epsilon)$ the corresponding block error probability are asymptotically such that

$$\lim_{n\to+\infty} n\tilde{\mathrm{P}}^{E}_{\mathrm{b},s_{\min},\gamma}(n,\lambda(x),\rho(x),\epsilon) = \frac{1}{2}\frac{\mu\epsilon}{1-\mu\epsilon} - \sum_{s=1}^{s_{\min}-1} s\mu_s\epsilon^s = \frac{1}{2}\frac{\mu\epsilon}{1-\mu\epsilon} - \frac{1}{2}\sum_{s=1}^{s_{\min}-1}(\mu\epsilon)^s, \quad (3.5)$$

$$\lim_{n\to+\infty} \tilde{\mathrm{P}}^{E}_{\mathrm{B},s_{\min},\gamma}(n,\lambda(x),\rho(x),\epsilon) = 1 - e^{\sum_{s=1}^{s_{\min}-1}\frac{(\mu\epsilon)^s}{2s}}\sqrt{1-\mu\epsilon}. \quad (3.6)$$

For the ensembles we consider in Lemma 1, the threshold is not given by the stability condition and therefore $\mu = \lambda'(0)\rho'(1) < 1/\epsilon^*$. This means that the condition $\mu\epsilon < 1$ is verified at least up to the threshold.

For ensemble with minimum variable degree $\mathtt{l}_{\min} > 2$ ($\mu = 0$), it was also shown in [20] that the probability that there is a stopping set of size s in the graph decays like $O(n^{-s(\mathtt{l}_{\min}/2-1)})$. Therefore when $\mathtt{l}_{\min} > 2$, if we fix a size s, then asymptotically, there is no stopping set of this size remaining in any graph as the blocklength increases and the error floor therefore vanishes.

Are the approximations given in (3.5) and (3.6) and stemming from the asymptotic analysis sufficient for moderate or short length codes? We show an example in Fig. 3.2, where we see that there is quite a big difference between the asymptotic and the finite-length curves.

The difference is particularly large for ensembles with $\mathtt{l}_{\min} > 2$ where the asymptotic error floor contribution vanishes completely but moderate length codes nevertheless exhibit an error floor (see Fig. 3.1). This is explained by the fact that asymptotically, only stopping sets involving exclusively degree-two variable nodes have a non-vanishing probability. But for fixed lengths, stopping sets involving nodes of other degrees still occur.

3.4 Approximation of the Errorfloor Curve

Consider the ensemble LDPC$(n,\lambda(x),\rho(x))$. The approximation which we will use is based on the following two simple assumptions. First, we will assume that the numbers of minimal stopping sets follow a jointly Poisson distribution with independent components. This is motivated by the fact that, as we have just seen, for n tending to infinity this is indeed true. Of course for finite blocklengths this is only an approximation. Second, we will use for the average of the numbers (not necessarily minimal) of stopping sets A_s the expressions of (3.2). Construct the generating function $A(x) = \sum_{s\geq 0} A_s x^s$ and call $\tilde{A}(x) = \sum_{s\geq 1}\tilde{A}_s x^s$ the generating function where $\tilde{A}_s$ is the average of the number of minimal stopping sets of size s. The two generating functions are

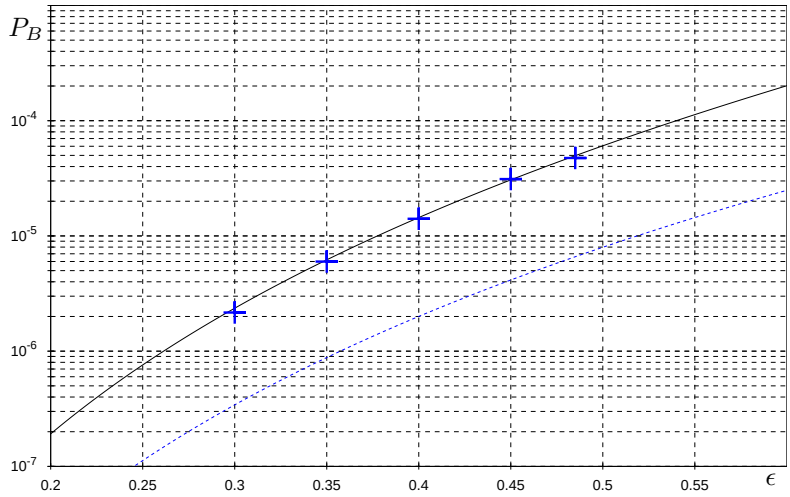


Figure 3.2: Error floor for the block error probability due to stopping sets of sizes between 6 and 26. The ensemble LDPC($n = 5000, \lambda(x) = 0.0739196x + 0.657891x^2 + 0.268189x^12, \rho(x) = 0.390753x^4 + 0.361589x^5 + 0.247658x^9$) is used over a binary erasure channel of erasure fraction ϵ. The dashed curve uses the asymptotic expression in (3.6), the solid line is our approximation of (3.8) and finally, the crosses are simulation points.

related in the following way

$$A(x) = e^{\tilde{A}(x)} = \sum_i \frac{\tilde{A}(x)^i}{i!}.$$

The motivation behind this formula is quite simple. The second term of the sum $\tilde{A}(x)$ counts the expected number of minimal stopping sets. The third term $\tilde{A}(x)^2/2$ adds to these all products of pairs and so on. This is true because (i) we have a distribution whose components are independent and therefore, the expectation of products of the components is the product of the expectations and (ii) two minimal stopping sets do not overlap with high probability so that the two weights of this union is equal to the sum of the two weights. This means that if we know the sequence A_s, we can to compute the sequence $\tilde{A}_s$.

$$\tilde{A}(x) = \sum_s \tilde{A}_s x^s = \log\left(A(x)\right).$$

This results in the following procedure to obtain an approximation for the error probability due to stopping sets of sizes $s \geq s_{\min}$. The different steps are:

1. Compute the expected number of stopping sizes of A_s for $s \geq 0$ according to (3.2) and

write the result in the generating function $A(x)$.

2. Compute $\log\left(A(x)\right)$ and define $\tilde{A}_s = \text{coef}\left\{\log\left(A(x)\right), x^s\right\}$ for $s \geq 1$.

3. Modify the approximations for the bit and the block error probability given (3.5) and (3.6) to obtain

$$\mathrm{P}^{E}_{\mathrm{b},s_{\min},\gamma}(n,\lambda(x),\rho(x),\epsilon) = \frac{1}{n}\sum_{s\geq s_{\min}} s\tilde{A}_s\epsilon^s \tag{3.7}$$

$$\mathrm{P}^{E}_{\mathrm{B},s_{\min},\gamma}(n,\lambda(x),\rho(x),\epsilon) = 1 - e^{-\sum_{s\geq s_{\min}}\tilde{A}_s\epsilon^s} \tag{3.8}$$

This approximation is shown in Fig. 3.1 for an ensemble with $\mathtt{l}_{\min} > 2$ and in Fig. 3.2 for an ensemble with $\mu \neq 0$.

Appendix

Efficient Evaluation of Powers of Polynomials

Evaluating coef$\{((1+x)^{\mathtt{r}} - \mathtt{r}x)^n, x^i\}$

Let $p(x) = \sum_i p_i x^i$ be a polynomial such that $p_0 \neq 0$ and assume that we want to compute the first coefficients of $q(x) = p(x)^n$. Taking the derivatives of $q(x)$ with respect to x results in

$$\begin{aligned} xq'(x)p(x) =& nxp'(x)q(x) \\ \sum_i \left(\sum_{k=0}^{i} p_k(i-k)q_{i-k} \right) x^i =& \sum_i n \left(\sum_{k=0}^{i} k p_k q_{i-k} \right) x^i, \end{aligned}$$

which gives

$$\begin{aligned} \sum_{k=0}^{i} p_k(i-k)q_{i-k} =& n \sum_{k=0}^{i} k p_k q_{i-k} \\ p_0 i q_i =& \sum_{k=1}^{i} (k(n+1)-i) p_k q_{i-k} \\ q_i =& \frac{1}{i p_0} \sum_{k=1}^{i} (k(n+1)-i) p_k q_{i-k}, \quad \forall i \neq 0. \end{aligned}$$

For $i = 0$, we simply have $q_0 = p_0^n$. This procedure enables us to compute the first j coefficients of $q(x)$ by performing a number of operations which is proportional to j but independent of n. If we take $p(x) = (1+x)^{\mathtt{r}} - \mathtt{r}x$ and therefore $q(x) = ((1+x)^{\mathtt{r}} - \mathtt{r}x)^n$, we obtain $q_0 = p_0^n = 1$ and $q_i = \frac{1}{i} \sum_{k=2}^{i} (k(n+1)-i) \binom{\mathtt{r}}{k} q_{i-k} = \frac{1}{i} \sum_{k=0}^{i-2} (n(i-k)-k) \binom{\mathtt{r}}{i-k} q_k$ □

Evaluating coef$\{(1+xy^{\mathtt{l}})^n, x^i y^{\mathtt{l}i}\}$

Consider now the polynomial $q(x) = (1+xy^{\mathtt{l}})^n = \sum_{k=0}^{n} q_k x^k y^{\mathtt{l}k}$ with $q_k = \binom{n}{k}$. By the definition of the binomial, we have $q_0 = 1$ and $q_i = \frac{n-i+1}{i} q_{i-1}$ for $1 \leq i \leq n$. This shows that we can evaluate the first j coefficients of $q(x)$ by performing a number of operations which is also only proportional to j. □

Chapter 4

Optimization

In this chapter, we will be concerned with the optimization of LDPC codes. Our approach will be to consider ensembles of codes LDPC$(n, \lambda(x), \rho(x))$ as defined and studied in the previous chapters and optimize the degree distribution. In other words, we will fix a channel and find degree distributions $\lambda(x)$, $\rho(x)$, such that transmission using random elements from this ensemble is successful with high probability and such that the rate of the code is maximal. In order to limit the dimensionality of the search space, we will set a maximum variable node degree $\mathtt{l}_{\max}$ as well as a maximum check node degree $\mathtt{r}_{\max}$.

In the first part of this chapter, we will consider such an optimization for infinite blocklengths. In that case, for the decoding to be successful, we need the threshold of the code to be higher than the channel parameter we are considering. We will start by reviewing how this type of optimization can be performed on the BEC and derive from there an optimization technique useful on any binary-input memoryless symmetric channel.

In the second part of this chapter, we will deal with the optimization for finite-length codes. In this case, our constraint will be to meet a target bit or block error probability for a fixed length.

4.1 Asymptotic Optimization

Luby et al. in [7] proposed to use Linear Programming to optimize the degree distributions. We will recall their approach and present how this optimization is done in the case of the BEC to set the framework for our optimization for a wider class of channels.

4.1.1 Binary Erasure Channel

Assume that we want to find degree distribution pairs $\lambda(x)$ and $\rho(x)$ of maximum rate such that the BP threshold is $\epsilon^* \geq 0.5$. We saw in Section 2.2 that the decoding is asymptotically successful for a channel erasure probability ϵ if $r_1(y) > 0$, $\forall y \in (0, 1]$. This constraint can be written as

$$\rho(1 - \epsilon\lambda(y)) > 1 - y, \qquad \forall y \in (0, 1] \tag{4.1}$$

or equivalently as

$$\epsilon\lambda(1 - \rho(1 - x)) < x, \qquad \forall x \in (0, \epsilon] \tag{4.2}$$

Rewriting (4.2) as

$$\sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i \epsilon (1 - \rho(1 - x))^{i-1} < x, \qquad \forall x \in (0, \epsilon], \tag{4.3}$$

we see that if the check node degree distribution $\rho(x)$ is fixed, the constraint becomes linear in terms of the λ_i. Also,

$$\begin{aligned} \sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i =& 1, \\ \lambda_i \geq& 0, \qquad \forall i \in \{1, \cdots, \mathtt{l}_{\max}\}, \end{aligned}$$

as the λ_i are positive fractions that should sum to one.

Our objective is to maximize the rate. If we assume that $\rho(x)$ is fixed, this is also a linear function of the λ_i,

$$\begin{aligned} \text{argmax}_{\lambda_1, \cdots, \lambda_{\mathtt{l}_{\max}}} rate =& \text{argmax}_{\lambda_1, \cdots, \lambda_{\mathtt{l}_{\max}}} 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} \\ =& \text{argmax}_{\lambda_1, \cdots, \lambda_{\mathtt{l}_{\max}}} 1 - \frac{\int_0^1 \rho(x)dx}{\sum_{i=1}^{\mathtt{l}_{\max}} \frac{\lambda_i}{i}} \\ =& \text{argmax}_{\lambda_1, \cdots, \lambda_{\mathtt{l}_{\max}}} \sum_{i=1}^{\mathtt{l}_{\max}} \frac{\lambda_i}{i}. \end{aligned}$$

Linear Program

To summarize, if one fixes $\rho(x)$ then finding the best $\lambda(x)$ amounts to solving the following Linear Program:

- **Constraints**
  - $\lambda_i \geq 0, \qquad \forall i \in \{1, \cdots, \mathtt{l}_{\max}\},$
  - $\sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i = 1,$
  - $\sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i \epsilon (1-\rho(1-x))^{i-1} < x, \qquad \forall x \in (0, \epsilon].$
- **Objective function to maximize**
  - $\sum_{i=1}^{\mathtt{l}_{\max}} \frac{\lambda_i}{i}$

One can then take the solution $\lambda(x)$, fix it and optimize in a similar way the check node degree distribution $\rho(x)$ using the linear constraints given this time by (4.1). Alternating between optimization of $\lambda(x)$ and $\rho(x)$ will give us better and better ensembles (having higher rates). Examples of ensembles optimized in this manner can be found in [34].

Alternative Representation

In order to find an optimization technique for general binary-input memoryless symmetric channels, we will start by representing differently the above optimization for the BEC. This alternative description is reminiscent of the EXIT chart method introduced by ten Brink in [35]. Consider again the case where $\rho(x)$ is fixed and we optimize $\lambda(x)$. The decoding is asymptotically successful if

$$\begin{aligned} \rho(1-\epsilon\lambda(y)) &> 1-y \qquad \forall y \in (0,1] \\ 1-\epsilon\lambda(y) &> \rho^{-1}(1-y) \qquad \forall y \in (0,1]. \end{aligned} \tag{4.4}$$

$\rho^{-1}(x)$ is well defined as $\rho(x)$ is a strictly increasing function in the interval $[0,1]$. Now, if we define the function $v(x)$ and for each $i \in \{1, \cdots, \mathtt{l}_{\max}\}$, the functions $v_i(x)$ such that

$$\begin{aligned} v_i(x) &= 1-\epsilon(1-x)^{i-1} \\ v(x) &= \sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i v_i(x) \\ &= 1-\epsilon\lambda(1-x) \end{aligned}$$

and the function $c(x)$ to be

$$c(x) = \rho(x)$$

The condition (4.4) translate to

$$v(x) > c^{-1}(x) \qquad \forall x \in [0,1) \tag{4.5}$$

$$\sum_{i=1}^{l_{\max}} \lambda_i v_i(x) > c^{-1}(x) \qquad \forall x \in [0,1) \tag{4.6}$$

Now, we can follow the same approach as in [35] and represent this constraint graphically as shown in Fig. 4.1. The curve of $v(x)$ has to be above the curve of $c^{-1}(x)$ for all $x \in [0,1)$. This

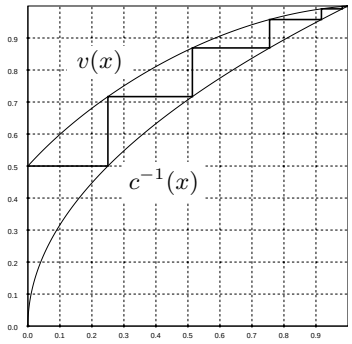


Figure 4.1: Curves of $v(x)$ and $c^{-1}(x)$ for $\lambda(x) = 0.15x + 0.8x^2 + 0.05x^9$ and $\rho(x) = x^2$. The decoder is successful as $v(x) > c^{-1}(x)$, $\forall x \in [0,1)$. The rate is 0.0384615

condition is verified for the code shown. The optimization proceeds by finding linear combinations of the curves $v_i(x)$ such that the rate is maximized and $v(x)$ is above $c^{-1}(x)$. As the optimization proceeds, the curves get closer and closer as explained in [36] and shown in Fig. 4.2.

Interpretation of the Curves $v(.)$ and $c(.)$

Consider the BP decoder after l iterations for an infinite graph having degree distributions $\lambda(x)$ and $\rho(x)$. Call y_l the fraction of erased messages sent from the check nodes to the variable nodes and $x_l = \epsilon\lambda(y_l)$ the fraction of erased messages sent from the variable nodes to the check nodes. Consider the following random variables. $\overleftarrow{M}_l$ is the value of the message sent from a check node to a variable on an edge that we pick uniformly at random in the graph. $\overrightarrow{M}_l$ correspond to a similar experiment except that this time we look at the message sent from the variable node to the check node. X is the value of the bit that was transmitted for the variable node connected

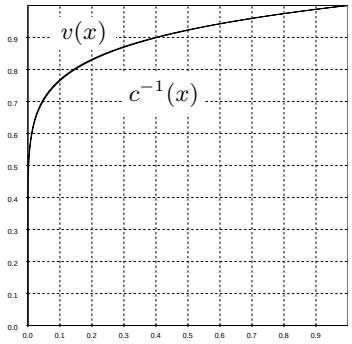


Figure 4.2: Curves of $v(x)$ and $c^{-1}(x)$ for $\lambda(x) = 0.226062x + 0.123125x^2 + 0.102154x^4 + 0.066579x^5 + 0.00172901x^{10} + 0.145591x^{11} + 0.0502078x^{28} + 0.0997158x^{29} + 0.184837x^{100}$ and $\rho(x) = 0.4x^8 + 0.6x^9$. The decoder is successful as $v(x) > c^{-1}(x)$, $\forall x \in [0, 1)$. The rate is 0.49.

to the edge. One can compute the mutual information [37, 35] between the messages and the transmitted bits to find

$$\begin{aligned} I(\overleftarrow{M}_l; X) =& 1 - y_l, \\ I(\overrightarrow{M}_l; X) =& 1 - x_l. \end{aligned}$$

This gives the following relationship between the mutual information of the messages incoming the variable nodes $I(\overleftarrow{M}_l; X)$ and the mutual information of the messages outgoing from the variable nodes $I(\overrightarrow{M}_l; X)$,

$$\begin{aligned} I(\overrightarrow{M}_l; X) =& 1 - x_l \\ =& 1 - \epsilon\lambda(y_l) \\ =& v(I(\overleftarrow{M}_l; X)). \end{aligned}$$

Therefore, the function $v(.)$ describes how the mutual information between the messages in the graph and the transmitted bits evolves when we process the variable nodes. The functions $v_i(.)$ are similar except that we only consider messages that are sent by variable nodes of degree $i \in \{1, \cdots, \}$.

The function $c(.)$ plays the equivalent role for the check node side. The staircase curve in Fig. 4.1 therefore represents how the mutual information between the messages and the transmitted bits evolves throughout the iterations. The decoder is successful if the mutual information converges to 1.

4.1.2 General Channels

Consider a binary-input memoryless symmetric channel parametrized by p. This can be for example an Additive White Gaussian Noise (AWGN) channel with standard deviation $p = \sigma$. Our objective is to construct in this case too, an efficient optimization procedure that relies on curve fitting using Linear Programming.

For a BP decoder at the l^{th} iteration. We can define similarly to the case of the BEC, the random variables $\overleftarrow{M}_l$ and $\overrightarrow{M}_l$ except that this time, they take values in $\mathbb{R}$. We are interested in the mutual information between $\overleftarrow{M}_l$ or $\overrightarrow{M}_l$ and X, the transmitted bit and we would like to find a function $v(.)$ such that

$$I(\overrightarrow{M}_l; X) = v(I(\overleftarrow{M}_l; X)).$$

In other words, we would like the function $v(.)$ to describe how the mutual information between the messages and the transmitted bits evolves when we process the variable nodes.

This is where the difficulties arise. Except for the case of the BEC, the mutual information of the messages that are sent by the variable nodes does not depend solely on the mutual information of the incoming messages. It also depends on the actual density of the incoming messages.

EXIT Chart Method

This issue was treated in [35] in the following way. As an approximation, ten Brink assumed that the densities of the incoming messages to the variable node belong to the family of Gaussian densities of which the mean is the double of the variance. This relationship between the mean and the variation is motivated by the fact that in the case of an AWGN channel of variance σ^2, the log-likelihood of the received values are distributed according to Gaussian of mean $2/\sigma^2$ and variance $4/\sigma^2$. For this particular family, there is a one-to-one correspondence between the mutual information and the density. Therefore, under this assumption, it is possible to follow the same procedure as for the BEC. Define functions $v_i(.)$ for each variable degree i, a function $c(.)$ for the check nodes and optimize for the rate with the constraint that $\sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i v_i(x) > c^{-1}(x)$, $\forall x \in [0, 1)$.

Let us review explicit expressions for the functions $v_i(.)$ and $c(.)$ when the channel is AWGN with a standard deviation σ. Without loss of generality, we assume that the all-one codeword is sent (see [7, 20]). In this case, the initial log-likelihood of the received values are distributed according to Gaussian of mean $2/\sigma^2$ and variance $4/\sigma^2$. Consider now the variable nodes of

degree i assuming that the input messages coming the check nodes are distributed according to a Gaussian belonging to the family defined above. Call $\psi(m)$, the functions that maps the mean m of such a Gaussian to the mutual information of the corresponding messages.

The functions $v_i()$, $i \in \{1, \cdots, \mathtt{l}_{\max}\}$ can be written as

$$v_i(x) = \psi((i-1)\psi^{-1}(x) + 2/\sigma^2).$$

Here, x is the mutual information of the incoming messages that are distributed according to a Gaussian of mean $\psi^{-1}(x)$. In each variable node of degree i, we sum $(i-1)$ such messages to which we add the initial log-likelihood which is a Gaussian of mean $2/\sigma^2$. The resulting outgoing message is a Gaussian with mean $(i-1)\psi^{-1}(x) + 2/\sigma^2$ and variance the double of this quantity. It corresponds therefore to a mutual information equal to $\psi((i-1)\psi^{-1}(x)+2/\sigma^2)$. For the check nodes, we can use the function

$$c(x) = 1 - \psi((i-1)\psi^{-1}(1-x)),$$

which makes use of the same assumption on the input densities as before and of an additional approximation due the duality principle introduced by Chung [38]. Now that we have the functions $v_i(.)$ and $c(.)$ defined, we can proceed in a manner identical to the one before and optimize the degree distribution and optimize the degree distributions using Linear Programming.

Although the EXIT chart method described above gives already satisfactory results, it is based on the assumption that the intermediate densities are Gaussian and it is therefore only an approximation. In Fig 4.3, we show as an example the densities of the messages sent from the variable nodes and from the check nodes of a BP decoder after 10 iterations. We see that the Gaussian assumption is questionable in this case.

Our Approach

Consider the following procedure. Start with the degree distributions $\lambda(x)$ and $\rho(x)$ and run density evolution [38, 12]. This gives a discrete set of incoming and outgoing densities for the variable nodes. One can easily compute the mutual information associated to each density, therefore obtaining the exact evolution of the mutual information during this decoding. This is depicted on the left side of Fig. 4.4. Now define a new function $v(x)$ by joining the discrete set of points showing the mutual information input output relation for the variable nodes and apply the same procedure for $c(x)$ and the check nodes. The resulting piecewise affine functions are

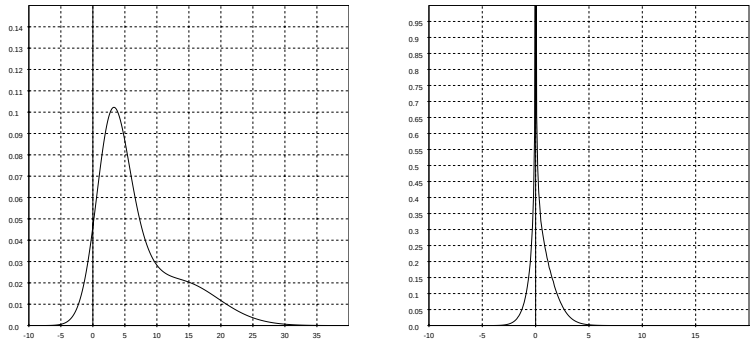


Figure 4.3: Densities after 10 iterations of BP for $\lambda(x) = 0.212332x + 0.197596x^2 + 0.0142733x^4 + 0.0744898x^5 + 0.0379457x^6 + 0.0693008x^7 + 0.086264x^8 + 0.00788586x^{10} + 0.0168657x^{11} + 0.283047x^{30}$ and $\rho(x) = x^8$. The channel is the AWGN with $\sigma = 0.93$. The threshold of the code is $\sigma^* = 0.9714$. The curve on the left is the density outgoing of variable node and the curve on the right is the density outgoing from check nodes, both conditioned on $X = +1$.

depicted on the right side of Fig. 4.4. Consider also the variable nodes of each degree separately and construct the functions $v_i(x)$, $\forall i \in \{1, \cdots, \mathtt{l}_{\max}\}$ in the same way.

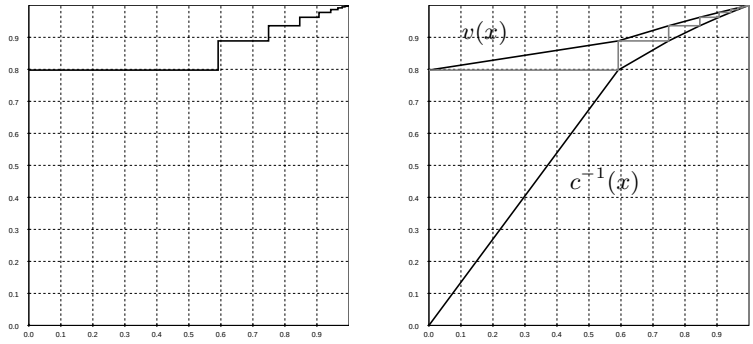


Figure 4.4: Evolution of the mutual information for the degree distribution $\lambda(x) = 0.405255x + 0.188018x^2 + 0.104114x^4 + 0.0333434x^5 + 0.108763x^8 + 0.160506x^{24}$ and $\rho(x) = 0.35x^2 + 0.65x^3$. The channel is AWGN with $\sigma = 1$.

The functions $v_i(x)$, $i \in \{1, \cdots, \mathtt{l}_{\max}\}$, $v(x)$ and $c(x)$ have the property that they describe exactly the evolution of the mutual information for this particular channel and degree distribution. Now if one varies slightly the degree distribution $\lambda(x)$, it is reasonable to assume that the intermediate densities of the messages in the new BP decoder remain "close" to the original ones. Assuming this is true, the functions $v_i(x)$ will be meaningful approximations to the relationship

between input and output mutual information for variable nodes of degree $i \in \{1, \cdots, \mathtt{l}_{\max}\}$. The equivalent statement applies to $c(x)$.

One can now use this set of functions to optimize the degree distribution $\lambda(x)$ by curve fitting using a Linear Program as described previously. However, this time we only allow small changes in $\lambda(x)$ in order to make sure that the close set of intermediate densities is meaningful.

This complete procedure is now repeated. At each round of the optimization, the functions $v_i(x)$ for $i \in \{1, \cdots, \mathtt{l}_{\max}\}$ and $c(x)$ are obtained through a piecewise affine interpolation between the points obtained from density evolution. Note that at those "spots" where the two curves are "close" to each other, we have many interpolation points and so an accurate approximation. On the other hand, if the curves are far apart, we only have few interpolation points, but in this case, an accurate representation of the curves is not necessary. Fig. 4.5 shows such an evolution of the mutual information for an optimized code. The decoder required ~ 1400 iterations of BP to converge.

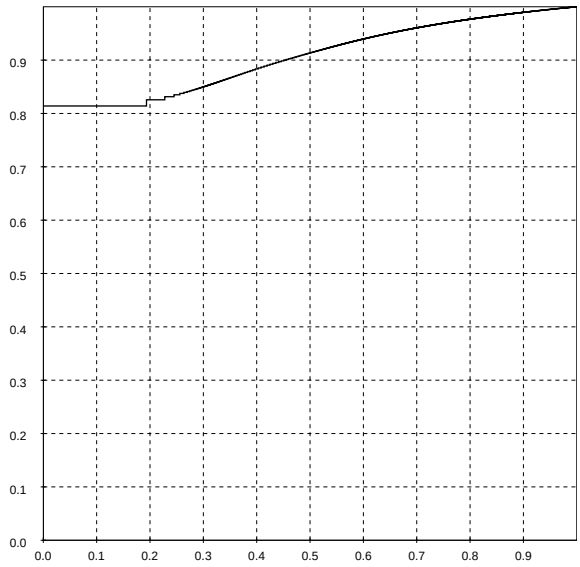


Figure 4.5: Evolution of the mutual information for the degree distribution $\lambda(x) = 0.161077x + 0.15439x^2 + 0.0455875x^5 + 0.152281x^6 + 0.100217x^{17} + 0.0138835x^{19} + 0.0761624x^{21} + 0.00598335x^{22} + 00.290418x^{99}$ and $\rho(x) = 0.5x^{10} + 0.5x^{11}$. The threshold of the code is $\sigma^* = 0.975347$ and its rate is 0.5. The channel we use here is AWGN with $\sigma = 0.974$.

This optimization procedure has been applied in [34] to optimize degree distributions for the AWGN. The optimized example shown in Fig. 4.5 is taken from [34]. In Fig. 4.6 we show the

gap to capacity for the codes optimized in [34] as a function of the rate.

The optimization procedure in [34] was performed each time for a fixed concentrated check node degree distribution (at most two consecutive degrees). Since the number of check node degrees for which an optimization was performed varies from rate to rate, the shape of the curve is somewhat rugged.

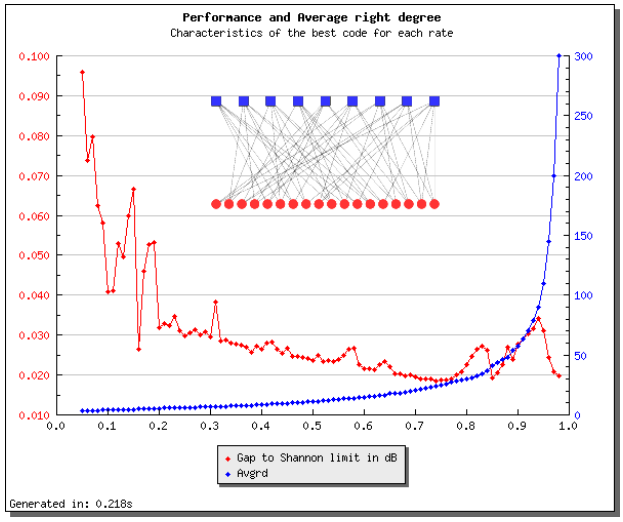


Figure 4.6: Gap to capacity for the codes optimized in [34] as a function of the rate.

This same optimization procedure has been applied also in [39] to show that optimized LDPC codes can approach the capacity of an AWGN multiple access channel without requiring time-sharing or rate-splitting [37, 40].

4.2 Finite-Length Optimization

Consider a BEC of erasure probability ϵ, a fixed blocklength n and a target block (or respectively bit) error probability P_{target}. We want to find degree distributions $\lambda(x)$ and $\rho(x)$ such that the average block (or respectively bit) error probability when using random elements from LDPC$(n, \lambda(x), \rho(x))$ on this channel is smaller than P_{target} while the rate of the code ensemble is maximized. We also consider the case of expurgated ensembles and in that case count only the probability of error stemming from failures composed of at least $s_{\min}$ bits. There are two possible

applications for this assumption. Either we use an outer code to "clean up" small errors, or we expurgate the ensemble and consider only the subset of codes that do not contain stopping sets of size smaller than $s_{\min}$. Our objective is to provide in the setting of LDPC codes an approach that is in principle applicable to a wide range of code families.

The two previous chapters have provided us with efficient approximations of the error probability curves in both the waterfall and the error floor regions. Combining these two approximations, results in an accurate assessment of the performance of the codes for any channel erasure probability. We show an example of this approximation in Fig. 4.7. For a fixed ϵ and n, we call $\mathrm{P}_B^W(\lambda,\rho)$ and $\mathrm{P}_b^W(\lambda,\rho)$ the approximation of the block and bit error probability in the waterfall region obtained in Section 2.6. Similarly, call $\mathrm{P}_B^E(\lambda,\rho)$ and $\mathrm{P}_b^E(\lambda,\rho)$ the approximation for the block and bit error probability in the error floor region given in Section 3.4. We drop the arguments n and ϵ to simplify the notation. The overall approximation of the probability of block error is $P_B(\lambda,\rho) = P_B^W(\lambda,\rho) + P_B^E(\lambda,\rho)$ and similarly it is $P_b(\lambda,\rho) = P_b^W(\lambda,\rho) + P_b^E(\lambda,\rho)$ for bit error probability.

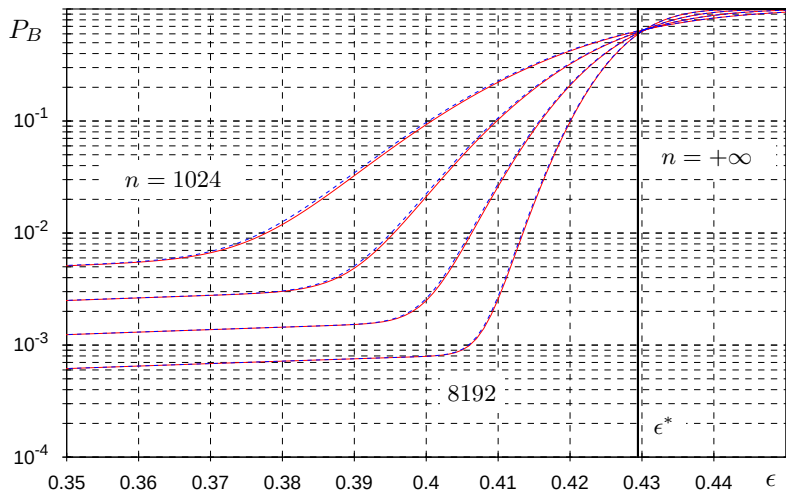


Figure 4.7: Block error probability curves for LDPC$(n, \lambda(x) = x^2, \rho(x) = x^5)$ when used over a BEC of erasure probability ϵ. The different curves are for $n \in \{1024, 2048, 4096, 8192\}$. The threshold is $\epsilon^* = 0.42943981$. The solid curves are the exact block error probabilities and the dashed curves our approximation.

We use these approximation to optimize the degree distributions for either a target block error probability or a target bit error probability.

Our approach is the following. $P_B(\lambda,\rho)$ and similarly $P_b(\lambda,\rho)$ are well defined functions of the degree distributions $\lambda(x)$ and $\rho(x)$, and we compute their total derivative with respect to the λ_i, and the ρ_j with $i \in \{1,\cdots,\mathtt{l}_{\max}\}$ and $j \in \{1\cdots\mathtt{r}_{\max}\}$. We can then use these derivatives to optimize the degree distributions. We will show how one can compute this total derivative in Section 4.2.2, but let us explain first the procedure in a generic way. This generic procedure is identical in both the optimization with respect to a block or to a bit target error probability. We will therefore outline it only in the case of the block target error probability.

Assume we change slightly the degree distributions to $\lambda(x)+\Delta\lambda(x)$ and $\rho(x)+\Delta\rho(x)$, then

$$\begin{aligned}\mathrm{P}_B(\lambda+\Delta\lambda,\rho+\Delta\rho) &\simeq \mathrm{P}_B(\lambda,\rho)+\Delta\,\mathrm{P}_B(\lambda,\rho)\\ &\simeq \mathrm{P}_B(\lambda,\rho)+\sum_{i=1}^{\mathtt{l}_{\max}}\Delta\lambda_i\frac{\partial\,\mathrm{P}_B}{\partial\lambda_i}+\sum_{i=1}^{\mathtt{r}_{\max}}\Delta\rho_i\frac{\partial\,\mathrm{P}_B}{\partial\rho_i}.\end{aligned}$$

The change of the probability of error $\Delta\,\mathrm{P}_B(\lambda,\rho)$ is expressed as a linear function of $\Delta\lambda_i$ and $\Delta\rho_j$, for $i\in\{1,\cdots,\mathtt{l}_{\max}\}$ and $j\in\{1,\cdots,\mathtt{r}_{\max}\}$. Similarly, the change of the rate of the code can also be written as

$$\begin{aligned}\Delta rate &\simeq \sum_{i=1}^{\mathtt{l}_{\max}}\Delta\lambda_i\frac{\partial rate}{\partial\lambda_i}+\sum_{i=1}^{\mathtt{r}_{\max}}\Delta\rho_i\frac{\partial rate}{\partial\rho_i}\\ &\simeq \sum_{i=1}^{\mathtt{l}_{\max}}\Delta\lambda_i\frac{\Lambda'(1)(1-rate)}{i}-\sum_{i=1}^{\mathtt{r}_{\max}}\Delta\rho_i\frac{\Lambda'(1)}{i}.\end{aligned}$$

Finally, we need the new distributions to be valid degree distributions and to be close to the original ones. Therefore, choose a "small" non-negative δ and require that

$$\begin{aligned}\sum_{i=1}^{\mathtt{l}_{\max}}\Delta\lambda_i &= 0,\\ \max\{-\lambda_i,-\delta\} \le \Delta\lambda_i &\le \min\{\delta,1-\lambda_i\},\qquad \forall i\in\{1,\cdots,\mathtt{l}_{\max}\}\end{aligned}$$

and

$$\begin{aligned}\sum_{i=1}^{\mathtt{r}_{\max}}\Delta\rho_i &= 0,\\ \max\{-\rho_i,-\delta\} \le \Delta\rho_i &\le \min\{\delta,1-\rho_i\},\qquad \forall i\in\{1,\cdots,\mathtt{r}_{\max}\}.\end{aligned}$$

This leads to the following optimization using Linear Programming.

4.2.1 Optimization Algorithm

Let us describe how the optimization proceeds. We have the following constraints

- The channel erasure probability is ϵ.
- The blocklength is n.
- The maximum variable degree is $\mathtt{l}_{\max}$ and the maximum check degree is $\mathtt{r}_{\max}$.
- We consider expurgated ensembles and count only errors larger or equal to $s_{\min}$ bits.
- The target block error probability is P_{target}.

Choose an initial degree distributions $\lambda(x)$ and $\rho(x)$. If $\mathrm{P}_B(\lambda,\rho) > P_{\text{target}}$, we need to decrease the block error probability of our ensemble. This is done by performing several rounds of the following Linear Program.

Linear Program to Decrease $P_B(\lambda,\rho)$.

- **Constraints**
  - $\sum_{i=1}^{\mathtt{l}_{\max}} \Delta\lambda_i = 0$
  - $\sum_{i=1}^{\mathtt{r}_{\max}} \Delta\rho_i = 0$
  - $\max\{-\lambda_i, -\delta\} \le \Delta\lambda_i \le \min\{\delta, 1-\lambda_i\} \qquad \forall i \in \{1,\cdots,\mathtt{l}_{\max}\}$
  - $\max\{-\rho_i, -\delta\} \le \Delta\rho_i \le \min\{\delta, 1-\rho_i\} \qquad \forall i \in \{1,\cdots,\mathtt{r}_{\max}\}$
- **Objective function to minimize $\Delta\,\mathrm{P}_B(\lambda,\rho)$**
  - $\sum_{i=1}^{\mathtt{l}_{\max}} \Delta\lambda_i \frac{\partial \mathrm{P}_B}{\partial \lambda_i} + \sum_{i=1}^{\mathtt{r}_{\max}} \Delta\rho_i \frac{\partial \mathrm{P}_B}{\partial \rho_i}$.

After each round, set the new degree distributions to be $\lambda(x) = \lambda(x) + \Delta\lambda(x)$ and $\rho(x) = \rho(x) + \Delta\rho(x)$. The stepsize δ is adjusted dynamically between the rounds. The optimization rounds continue until $\mathrm{P}_B(\lambda,\rho) < P_{\text{target}}$.

Once we are below the target block error probability, we can start to optimize the rate. This time, we have an additional constraint to specify that the block error probability has to remain below P_{target}, which can be written as $\Delta\,\mathrm{P}_B(\lambda,\rho) < P_{\text{target}} - \mathrm{P}_B(\lambda,\rho)$. The optimization of the rate is done through several rounds of the following Linear Program.

Linear Program to Increase the Rate.

- **Constraints**

- $\sum_{i=1}^{\mathtt{l}_{\max}} \Delta\lambda_i = 0$
- $\sum_{i=1}^{\mathtt{r}_{\max}} \Delta\rho_i = 0$
- $\max\{-\lambda_i, -\delta\} \leq \Delta\lambda_i \leq \min\{\delta, 1-\lambda_i\} \qquad \forall i \in \{1, \cdots, \mathtt{l}_{\max}\}$
- $\max\{-\rho_i, -\delta\} \leq \Delta\rho_i \leq \min\{\delta, 1-\rho_i\} \qquad \forall i \in \{1, \cdots, \mathtt{r}_{\max}\}$
- $\sum_{i=1}^{\mathtt{l}_{\max}} \Delta\lambda_i \frac{\partial \mathrm{P}_B}{\partial \lambda_i} + \sum_{i=1}^{\mathtt{r}_{\max}} \Delta\rho_i \frac{\partial \mathrm{P}_B}{\partial \rho_i} < P_{\text{target}} - \mathrm{P}_B(\lambda, \rho)$

- **Objective function to maximize $\Delta rate$.**
  - $\sum_{i=1}^{\mathtt{l}_{\max}} \Delta\lambda_i \frac{\Lambda'(1)(1-rate)}{i} - \sum_{i=1}^{\mathtt{r}_{\max}} \Delta\rho_i \frac{\Lambda'(1)}{i}$.

After each round, set the new degree distributions to be $\lambda(x) = \lambda(x) + \Delta\lambda(x)$ and $\rho(x) = \rho(x) + \Delta\rho(x)$. The stepsize δ is adjusted dynamically between the rounds. The optimization rounds continue until no further progress is made.

This algorithm is not guaranteed to converge to a global maximum of the rate. However, as the function $\mathrm{P}_B(\lambda, \rho)$ is simple to evaluate, one can start this procedure with several different initial conditions and choose the best outcome. In practice, we observe that in fact the algorithm seems to converge to the same point for several different initial conditions.

4.2.2 Total Derivative

In this section, we will explain how we can compute the total derivative of the approximations of the error probability with respect to the degree distributions.

We consider an ensemble LDPC$(n, \lambda(x), \rho(x))$, a binary erasure channel of erasure probability ϵ.

Our approximation for the block and bit error probabilities are respectively

$$\begin{aligned} P_B(\lambda, \rho) =& P_B^W(\lambda, \rho) + P_B^E(\lambda, \rho), \\ P_b(\lambda, \rho) =& P_b^W(\lambda, \rho) + P_b^E(\lambda, \rho). \end{aligned}$$

where $P_B^W(\lambda, \rho)$ and $P_b^W(\lambda, \rho)$ are computed from the degree distributions as explained in Section 2.6 and where $P_B^E(\lambda, \rho)$ and $P_b^E(\lambda, \rho)$ are computed as explained in Section 3.4. Let us consider both contributions separately.

Total Derivative of $P_B^W(\lambda,\rho)$ and $P_b^W(\lambda,\rho)$

In Section 2.6, we define the approximations of the error probabilities for the waterfall region as

$$\begin{aligned}\mathrm{P_B^W}(\lambda,\rho) =& Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right),\\ \mathrm{P_b^W}(\lambda,\rho) =& \nu^* Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right).\end{aligned}$$

with

$$\begin{aligned}\alpha =& \left(\frac{\rho(\bar{x}^*)^2 - \rho(\bar{x}^{*2}) + \rho'(\bar{x}^*)(1-2x^*\rho(\bar{x}^*)) - \bar{x}^{*2}\rho'(\bar{x}^{*2})}{\Lambda'(1)\lambda(y^*)^2\rho'(\bar{x}^*)^2} + \right.\\ & \left.\frac{\epsilon^{*2}\lambda(y^*)^2 - \epsilon^{*2}\lambda(y^{*2}) - y^{*2}\epsilon^{*2}\lambda'(y^{*2})}{\Lambda'(1)\lambda(y^*)^2}\right)^{1/2},\\ \beta =& \left(\frac{\epsilon^{*4}r_2^{*2}(\epsilon^*\lambda'(y^*)^2 r_2^* - x^*(\lambda''(y^*)r_2^* + \lambda'(y^*)x^*))^2}{\Lambda'(1)^2\rho'(\bar{x}^*)^3 x^{*10}(2\epsilon^*\lambda'(y^*)^2 r_3^* - \lambda''(y^*)r_2^* x^*)}\right)^{1/3}\end{aligned}$$

and for $i \geq 2$

$$r_i^* = \sum_{m\geq j\geq i}(-1)^{i+j}\binom{j-1}{i-1}\binom{m-1}{j-1}\rho_m(\epsilon^*\lambda(y^*))^j.$$

We want to compute all the derivatives $\frac{\partial P_B^W(\lambda,\rho)}{\partial\lambda_i}$ and $\frac{\partial P_b^W(\lambda,\rho)}{\partial\lambda_i}$ for $i \in \{1,\cdots,\mathtt{l}_{\max}\}$ and $\frac{\partial P_B^W(\lambda,\rho)}{\partial\rho_i}$ and $\frac{\partial P_b^W(\lambda,\rho)}{\partial\rho_i}$ for $i \in \{1,\cdots,\mathtt{r}_{\max}\}$.

We make intensive use of the chain rule. We start by writing

$$\begin{aligned}dP_B^W(\lambda,\rho) =& \left(\frac{\sqrt{n}\left(d\epsilon^* - \frac{d\beta}{n^{2/3}}\right)}{\alpha} - \frac{\left(-\frac{\beta}{n^{2/3}} + \epsilon^* - \epsilon\right)\sqrt{n}d\alpha}{\alpha^2}\right)Q'\left(\frac{\left(-\frac{\beta}{n^{2/3}} + \epsilon^* - \epsilon\right)\sqrt{n}}{\alpha}\right),\\ dP_b^W(\lambda,\rho) =& \nu^* dP_B^W(\lambda,\rho) + P_B^W(\lambda,\rho)d\nu^*.\end{aligned}$$

We see that we now need to compute $d\epsilon^*$, $d\alpha$, $d\beta$ and $d\nu^*$. Expanding $d\beta$ as a function of $d\lambda_i$, $d\rho_j$, $d\epsilon^*$, dx^* and dy^* is best done using a mathematical software capable of symbolic calculations. We will therefore just write this expansion in a formal way.

Define the function $\beta_f(.)$ such that

$$\beta_f(\lambda_1,\cdots,\lambda_{\mathtt{l}_{\max}},\rho_1,\cdots,\rho_{\mathtt{r}_{\max}},\epsilon,x,y,r_2,r_3) = \left(\frac{\epsilon^4 r_2^2(\epsilon\lambda'(y)^2 r_2 - x(\lambda''(y)r_2 + \lambda'(y)x))^2}{\Lambda'(1)^2\rho'(\bar{x})^3 x^{10}(2\epsilon\lambda'(y)^2 r_3 - \lambda''(y)r_2 x)}\right)^{1/3}$$

The function $\beta_f(.)$ differs from β in the sense that it can be evaluated for any ϵ, x, y, r_2 and r_3 and not necessarily at the critical point corresponding to the degree distribution (which is the

case for β that is only a function of the degree distribution). Hence, using $\beta_f(.)$ and dropping the arguments in the expression, we can write

$$d\beta = \sum_{i=1}^{\mathtt{l}_{\max}} \frac{\partial \beta_f}{\partial \lambda_i} d\lambda_i + \sum_{i=1}^{\mathtt{r}_{\max}} \frac{\partial \beta_f}{\partial \rho_i} d\rho_i + \frac{\partial \beta_f}{\partial \epsilon} d\epsilon^* + \frac{\partial \beta_f}{\partial x} dx^* + \frac{\partial \beta_f}{\partial y} dy^* + \frac{\partial \beta_f}{\partial y} dr_2^* + \frac{\partial \beta_f}{\partial y} dr_3^*.$$

The partial derivatives of $\beta_f(.)$ can be computed easily.

Similarly for α, one can define the corresponding function $\alpha_f(.)$ and write

$$d\alpha = \sum_{i=1}^{\mathtt{l}_{\max}} \frac{\partial \alpha_f}{\partial \lambda_i} d\lambda_i + \sum_{i=1}^{\mathtt{r}_{\max}} \frac{\partial \alpha_f}{\partial \rho_i} d\rho_i + \frac{\partial \alpha_f}{\partial \epsilon} d\epsilon^* + \frac{\partial \alpha_f}{\partial x} dx^* + \frac{\partial \alpha_f}{\partial y} dy^*.$$

As we are progressing, we see that remains only to expand the partial derivatives $d\epsilon^*$, dy^*, dx^*, $d\nu^*$, dr_2^* and dr_3^*. Notice that again x^*, ν^*, r_2^* and r_3^* are expressed easily as a function of the degree distributions and of ϵ^* and y^*. Therefore, we can apply the same procedure as for α and β to obtain

$$\begin{aligned}
dx^* &= \sum_{i=1}^{\mathtt{l}_{\max}} \epsilon^* y^{*(i-1)} d\lambda_i + \lambda(y^*) d\epsilon^* + \epsilon \lambda'(y) dy^*, \\
d\nu^* &= \sum_{i=1}^{\mathtt{l}_{\max}} \frac{\partial \nu_f}{\partial \lambda_i} d\lambda_i + \frac{\partial \nu_f}{\partial \epsilon} d\epsilon^* + \frac{\partial \alpha_f}{\partial y} dy^*, \\
dr_2^* &= \sum_{i=1}^{\mathtt{l}_{\max}} \frac{\partial r_{2f}}{\partial \lambda_i} d\lambda_i + \sum_{i=1}^{\mathtt{r}_{\max}} \frac{\partial r_{2f}}{\partial \rho_i} d\rho_i + \frac{\partial r_{2f}}{\partial \epsilon} d\epsilon^* + \frac{\partial \alpha_f}{\partial y} dy^*, \\
dr_3^* &= \sum_{i=1}^{\mathtt{l}_{\max}} \frac{\partial r_{3f}}{\partial \lambda_i} d\lambda_i + \sum_{i=1}^{\mathtt{r}_{\max}} \frac{\partial r_{3f}}{\partial \rho_i} d\rho_i + \frac{\partial r_{3f}}{\partial \epsilon} d\epsilon^* + \frac{\partial \alpha_f}{\partial y} dy^*,
\end{aligned}$$

where we have expanded the calculations just for dx^* and kept the rest in generic form. The last step consist in computing the derivatives of ϵ^* and y^* with respect to the λ_i, $i \in \{1, \cdots, \mathtt{l}_{\max}\}$ and the ρ_i, $i \in \{1, \cdots, \mathtt{r}_{\max}\}$.

Recall the definition of ϵ^* and y^*. The threshold of the ϵ^* is the smallest ϵ such that the equation $r_1(y) = 0$ has a non-zero solution and y^* is this solution (see Section 2.2). As a consequence, at the critical point, we have $r_1(y^*) = 0$, but also $r_1'(y^*) = 0$. The curve touches the zero-line with a zero-derivative.

Using this property and defining the function $r_{1f}(.)$ that has as arguments ϵ, y but also of the degree distributions

$$r_1(\lambda_1, \cdots, \lambda_{\mathtt{l}_{\max}}, \rho_1, \cdots, \rho_{\mathtt{r}_{\max}}, \epsilon, y) = \epsilon \lambda(y)[y - 1 + \rho(1 - \epsilon \lambda(y))],$$

we obtain through the implicit function theorem that

$$\frac{\partial \epsilon^*}{\partial \lambda_i} = - \frac{\begin{vmatrix} \frac{\partial r_{1f}}{\partial y} & \frac{\partial r_{1f}}{\partial \lambda_i} \\ \frac{\partial^2 r_{1f}}{\partial y^2} & \frac{\partial^2 r_{1f}}{\partial y \partial \lambda_i} \end{vmatrix}}{\begin{vmatrix} \frac{\partial r_{1f}}{\partial y} & \frac{\partial r_{1f}}{\partial \epsilon} \\ \frac{\partial^2 r_{1f}}{\partial y^2} & \frac{\partial^2 r_{1f}}{\partial y \partial \epsilon} \end{vmatrix}}, \qquad \forall i \in \{1, \cdots, \mathtt{l}_{\max}\},$$

$$\frac{\partial \epsilon^*}{\partial \rho_i} = - \frac{\begin{vmatrix} \frac{\partial r_{1f}}{\partial y} & \frac{\partial r_{1f}}{\partial \rho_i} \\ \frac{\partial^2 r_{1f}}{\partial y^2} & \frac{\partial^2 r_{1f}}{\partial y \partial \rho_i} \end{vmatrix}}{\begin{vmatrix} \frac{\partial r_{1f}}{\partial y} & \frac{\partial r_{1f}}{\partial \epsilon} \\ \frac{\partial^2 r_{1f}}{\partial y^2} & \frac{\partial^2 r_{1f}}{\partial y \partial \epsilon} \end{vmatrix}}, \qquad \forall i \in \{1, \cdots, \mathtt{l}_{\max}\}$$

and similarly for y^*,

$$\frac{\partial y^*}{\partial \lambda_i} = - \frac{\begin{vmatrix} \frac{\partial r_{1f}}{\partial \epsilon} & \frac{\partial r_{1f}}{\partial \lambda_i} \\ \frac{\partial^2 r_{1f}}{\partial y \partial \epsilon} & \frac{\partial^2 r_{1f}}{\partial y \partial \lambda_i} \end{vmatrix}}{\begin{vmatrix} \frac{\partial r_{1f}}{\partial \epsilon} & \frac{\partial r_{1f}}{\partial y} \\ \frac{\partial^2 r_{1f}}{\partial y \partial \epsilon} & \frac{\partial^2 r_{1f}}{\partial y^2} \end{vmatrix}}, \qquad \forall i \in \{1, \cdots, \mathtt{l}_{\max}\},$$

$$\frac{\partial y^*}{\partial \rho_i} = - \frac{\begin{vmatrix} \frac{\partial r_{1f}}{\partial \epsilon} & \frac{\partial r_{1f}}{\partial \rho_i} \\ \frac{\partial^2 r_{1f}}{\partial y \partial \epsilon} & \frac{\partial^2 r_{1f}}{\partial y \partial \rho_i} \end{vmatrix}}{\begin{vmatrix} \frac{\partial r_{1f}}{\partial \epsilon} & \frac{\partial r_{1f}}{\partial y} \\ \frac{\partial^2 r_{1f}}{\partial y \partial \epsilon} & \frac{\partial^2 r_{1f}}{\partial y^2} \end{vmatrix}}, \qquad \forall i \in \{1, \cdots, \mathtt{l}_{\max}\},$$

Now we can recursively reconstruct the total derivative with respect to the degree distribution at all stages of our expansion. All functions have to be evaluated at the critical point. We obtain in this way, the expressions for the derivatives $\frac{\partial P_B^W(\lambda,\rho)}{\partial \lambda_i}$ and $\frac{\partial P_b^W(\lambda,\rho)}{\partial \lambda_i}$ for $i \in \{1, \cdots, \mathtt{l}_{\max}\}$ and $\frac{\partial P_B^W(\lambda,\rho)}{\partial \rho_i}$ and $\frac{\partial P_b^W(\lambda,\rho)}{\partial \rho_i}$ for $i \in \{1, \cdots, \mathtt{r}_{\max}\}$.

If we use the refined approximation for several critical points, the total derivative of each term (Q-function) is calculated in this way.

Total Derivative of $P_B^E(\lambda,\rho)$ and $P_b^E(\lambda,\rho)$

In Section 3.4, we define the approximations of the error probabilities in the error floor region as,

$$\begin{aligned}\mathrm{P_b^E}(\lambda,\rho) =& \frac{1}{n}\sum_{s\geq s_{\min}} s\tilde{A}_s\epsilon^s \\ \mathrm{P_B^E}(\lambda,\rho) =& 1-e^{-\sum_{s\geq s_{\min}}\tilde{A}_s\epsilon^s}\end{aligned}$$

The $\tilde{A}_s$ are obtained by the following procedure.

1. Compute the expected number of stopping sizes of A_s for $s\geq 1$ according to (3.2) and write the result in the generating function $A(x)$. We remind the expression of (3.2).

$$A_s = \sum_e \mathrm{coef}\left\{\prod_i (1+xy^i)^{n\Lambda_i}, x^s y^e\right\}\frac{\mathrm{coef}\left\{\prod_i((1+x)^i-ix)^{n_c P_i}, x^e\right\}}{\binom{n\Lambda'(1)}{e}}. \tag{4.7}$$

2. Compute $\log\left(A(x)\right)$ and define $\tilde{A}_s = \mathrm{coef}\left\{\log\left(A(x)\right), x^s\right\}$ for $s\geq 1$.

Since in these expressions, the degree distributions appear as integers (number of nodes of each type) it is more natural in this case to compute differences rather than derivatives.

In (4.7), the first term on the right hand side counts the number of possible sets of s variable nodes having e edges emanating from them. Therefore this term should be independent of check node distribution. Similarly, the second on the right is the probability that e edges connected to the check nodes give rise to a stopping. This is independent of the variable node degree. For this reason, one can consider each of the terms separately.

Let us start with the first term $\mathrm{coef}\left\{\prod_i(1+xy^i)^{n\Lambda_i}, x^s y^e\right\}$ and assume that we have already evaluated these coefficients for our degree distribution and for all possible possible s and e. Consider now a graph having one additional variable node of degree k. Then for the same pair e and s, we will need to evaluate $\mathrm{coef}\left\{\prod_i(1+xy^i)^{n\Lambda_i}(1+xy^k), x^s y^e\right\}$. However, one can easily see that this quantity can be obtained from values we already computed through the following relationship.

$$\begin{aligned}\mathrm{coef}\left\{\prod_i(1+xy^i)^{n\Lambda_i}(1+xy^k), x^s y^e\right\} =& \mathrm{coef}\left\{\prod_i(1+xy^i)^{n\Lambda_i}, x^s y^e\right\} \\ &+ \mathrm{coef}\left\{\prod_i(1+xy^i)^{n\Lambda_i}, x^{s-1}y^{e-k}\right\}\end{aligned}$$

Consider now the term $\frac{\text{coef}\left\{\prod_i((1+x)^i-ix)^{n_cP_i},x^e\right\}}{\binom{n\Lambda'(1)}{e}}$, and assume that we have already computed the coefficients for all e. If we add a check node of degree k to the set of check nodes, we have the following relationship that enables us to compute easily the new coefficients from the ones we already know.

$$\begin{aligned}&\frac{\text{coef}\left\{\prod_i((1+x)^i-ix)^{n_cP_i}((1+x)^k-kx),x^e\right\}}{\binom{n\Lambda'(1)+k}{e}}\\ =&\frac{\binom{n\Lambda'(1)}{e}}{\binom{n\Lambda'(1)+k}{e}}\left(\frac{\text{coef}\left\{\prod_i((1+x)^i-ix)^{n_cP_i},x^e\right\}}{\binom{n\Lambda'(1)}{e}}+\sum_{j=2}^{k}\binom{k}{j}\frac{\text{coef}\left\{\prod_i((1+x)^i-ix)^{n_cP_i},x^{e-j}\right\}}{\binom{n\Lambda'(1)}{e}}\right).\end{aligned}$$

Now that we know how to evaluate efficiently the new coefficients, apply the following procedure. For each $k\in\{1,\cdots,\mathtt{l}_{\max}\}$, compute $P_B^E(\lambda,\rho)$, then add a variable node of degree k, apply the whole procedure of (3.4) to obtain a new value of the approximation. Call the difference between the two values $\Delta_k^\lambda P_B^E(\lambda,\rho)$. Define similarly $\Delta_k^\lambda P_b^E(\lambda,\rho)$. For the check nodes the change of the value of the approximation after adding a check node of degree k is called $\Delta_k^\rho P_B^E(\lambda,\rho)$ for the block error approximation and $\Delta_k^\rho P_b^E(\lambda,\rho)$ for the bit error approximation. In order to relate these quantities to the derivative with respect to λ_i and ρ_j for $i\in\{1,\cdots,\mathtt{l}_{\max}\}$ and $j\in\{1,\cdots,\mathtt{r}_{\max}\}$. We have relate a change in these λ_i and ρ_j to the change of the number of nodes in the graph.

Call $\xi=n\Lambda'(1)=\frac{n}{\sum_{i=1}^{\mathtt{l}_{\max}}\frac{\lambda_i}{i}}$ the total number of edges in the graph. When λ_k varies by $\Delta\lambda_k$ the total number of edges in the graph varies such that,

$$\frac{\Delta\xi}{\Delta\lambda_k}=-\frac{n}{k\left(\sum\frac{\lambda_i}{i}\right)^2}.$$

In our definition of the ensembles LDPC$(n,\lambda(x),\rho(x))$ in Section 1.1, the number of edges in the graph depended solely on the variable node distribution, therefore, we will take

$$\frac{\Delta\xi}{\Delta\rho_k}=0.$$

The number of nodes variable nodes of degree $i\in\{1,\cdots,\mathtt{l}_{\max}\}$ and of check nodes of degree $j\in\{1,\cdots,\mathtt{r}_{\max}\}$ is written as

$$\begin{aligned}n\Lambda_i=&\frac{\xi\lambda_i}{i},\\ n_cP_j=&\frac{\xi\rho_j}{j}.\end{aligned}$$

Taking small variations in the edges perspective degree distributions gives

$$\begin{aligned}
\frac{\Delta n\Lambda_i}{\Delta\lambda_k} =& -n\frac{\lambda_i}{ik(\sum\frac{\lambda_j}{j})^2}, && \text{if } i \neq k,\\
\frac{\Delta n\Lambda_i}{\Delta\lambda_i} =& n\frac{(\sum\frac{\lambda_j}{j}), -\frac{\lambda_i}{i}}{i(\sum\frac{\lambda_j}{j})^2},\\
\frac{\Delta n_c P_j}{\Delta\rho_k} =& \frac{n}{k\sum\frac{\lambda_i}{i}}, && \text{if } i \neq k.
\end{aligned}$$

Finally, we obtain for $i \in \{1, \cdots, \mathtt{l}_{\max}\}$ and $j \in \{1, \cdots, \mathtt{r}_{\max}\}$

$$\begin{aligned}
\frac{\Delta \mathrm{P}_{\mathrm{B}}^E(\lambda,\rho)}{\Delta\lambda_i} &= \sum_{k=1}^{\mathtt{l}_{\max}} \Delta_k^\lambda P_B^E(\lambda,\rho)\frac{\Delta n\Lambda_k}{\Delta\lambda_i},\\
\frac{\Delta \mathrm{P}_{\mathrm{b}}^E(\lambda,\rho)}{\Delta\lambda_i} &= \sum_{k=1}^{\mathtt{l}_{\max}} \Delta_k^\lambda P_b^E(\lambda,\rho)\frac{\Delta n\Lambda_k}{\Delta\lambda_i},
\end{aligned}$$

$$\begin{aligned}
\frac{\Delta \mathrm{P}_{\mathrm{B}}^E(\lambda,\rho)}{\Delta\rho_j} &= \sum_{k=1}^{\mathtt{r}_{\max}} \Delta_k^\rho P_B^E(\lambda,\rho)\frac{\Delta n_c P_k}{\Delta\rho_j},\\
\frac{\Delta \mathrm{P}_{\mathrm{b}}^E(\lambda,\rho)}{\Delta\rho_j} &= \sum_{k=1}^{\mathtt{r}_{\max}} \Delta_k^\rho P_b^E(\lambda,\rho)\frac{\Delta n_c P_k}{\Delta\rho_j},
\end{aligned}$$

We will use these quantities as our derivatives for our approximations.

4.2.3 Sample Optimization

Consider the following constraints

- The channel erasure probability is $\epsilon = 0.5$.
- The blocklength is $n = 5000$.
- The maximum variable degree is $\mathtt{l}_{\max} = 13$ and the maximum check degree is $\mathtt{r}_{\max} = 10$.
- We count only errors larger or equal to $s_{\min} = 6$ bits.
- The target block error probability is $P_{\text{target}} = 10^{-4}$.

We start with randomly generated degree distributions

$$\begin{aligned}\lambda(x) =& 0.139976x + 0.149265x^2 + 0.174615x^3 + 0.110137x^4 + 0.0184844x^5 \\ &+ 0.0775212x^6 + 0.0166585x^7 + 0.00832646x^8 + 0.0760256x^9 \\ &+ 0.0838369x^{10} + 0.0833654x^{11} + 0.0617885x^{12}\end{aligned} \tag{4.8}$$

$$\begin{aligned}\rho(x) =& 0.0532687x + 0.0749403x^2 + 0.11504x^3 + 0.0511266x^4 + 0.170892x^5 \\ &+ 0.17678x^6 + 0.0444454x^7 + 0.152618x^8 + 0.160889x^9.\end{aligned} \tag{4.9}$$

The approximation of the block error probability curve of this code is shown in Fig. 4.8.

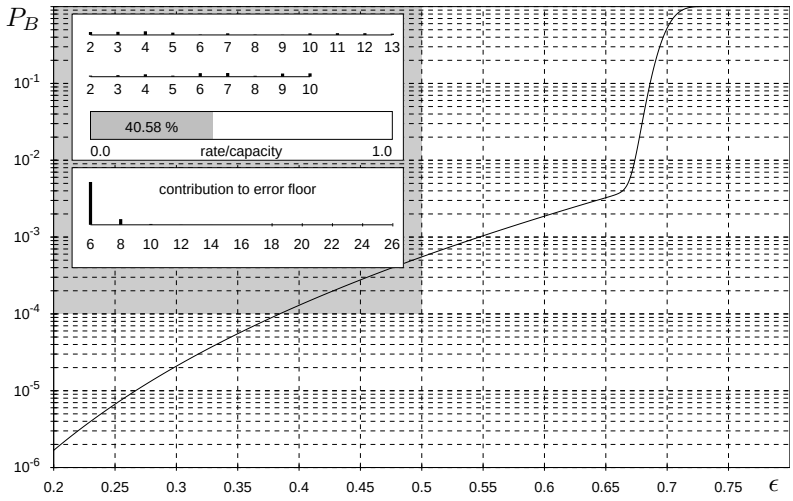


Figure 4.8: Approximation of the block error probability for the initial ensemble before the optimization (see (4.8) and (4.9)).

Let us explain briefly the format of the plot. The gray area represents our constraint on the target block error probability. An ensemble that fulfills the constraint $\mathrm{P}_B(\lambda, \rho) < P_{\text{target}}$ will therefore have a block error probability curve that does not cross the gray square. The first square starting from the top inside the gray area represents visually the degree distributions. The above lines represent respectively the histogram of the variable node and the check node distributions. Also in the same square, we visualize rate of the code by its ratio to the capacity of the channel of interest (here $\epsilon = 0.5$). The second square in the gray area shows the histogram of the contributions of the stopping sets of different sizes in the error floor at $\epsilon = 0.5$.

For this initial degree distribution, we have that the rate is 0.202922 and at $\epsilon = 0.5$, $\mathrm{P}_B(\lambda, \rho) = 0.000552 > P_{\text{target}}$. Therefore as explained previously, we have to start by minimizing $\mathrm{P}_B(\lambda, \rho)$

until it becomes lower that P_{target}.

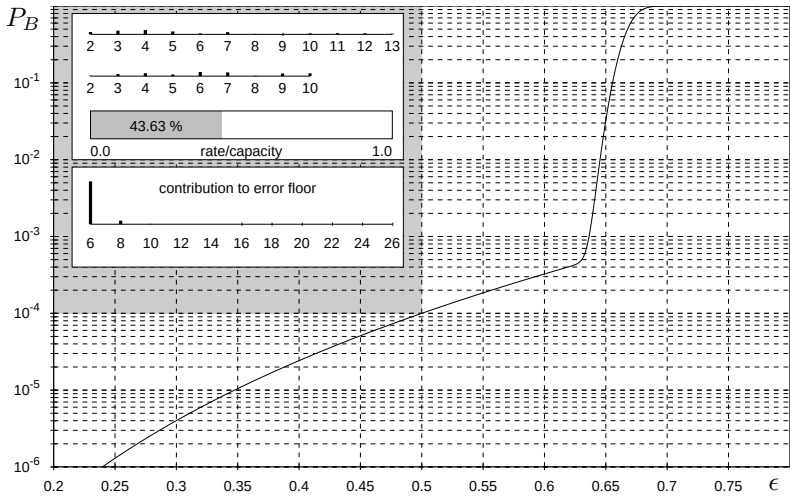


Figure 4.9: Approximation of the block error probability for the ensemble obtained after the first part of the optimization (see (4.10) and (4.11)). The error probability has been lowered below the target.

After a certain number of optimization rounds to decrease $\mathrm{P}_B(\lambda, \rho)$, we finally obtain the degree distributions

$$\begin{aligned}\lambda(x) =&0.111913x + 0.178291x^2 + 0.203641x^3 + 0.139163x^4 + 0.0475105x^5 + 0.106547x^6 \\ &+ 0.0240221x^7 + 0.0469994x^9 + 0.0548108x^{10} + 0.0543393x^{11} + 0.0327624x^{12}\end{aligned} \tag{4.10}$$

$$\begin{aligned}\rho(x) =&0.0242426x + 0.101914x^2 + 0.142014x^3 + 0.0781005x^4 + 0.198892x^5 \\ &+ 0.177806x^6 + 0.0174716x^7 + 0.125644x^8 + 0.133916x^9.\end{aligned} \tag{4.11}$$

that have $\mathrm{P}_B(\lambda, \rho) = 0.0000997$ and a rate of 0.21815. We show the corresponding approximation in Fig. 4.9.

Now, we start the second phase of the optimization and optimize the rate while insuring that the block error probability remains below the target. The resulting degree distribution is

$$\lambda(x) =0.0739196x + 0.657891x^2 + 0.268189x^{12} \tag{4.12}$$

$$\rho(x) =0.390753x^4 + 0.361589x^5 + 0.247658x^9 \tag{4.13}$$

It has rate 0.41065.

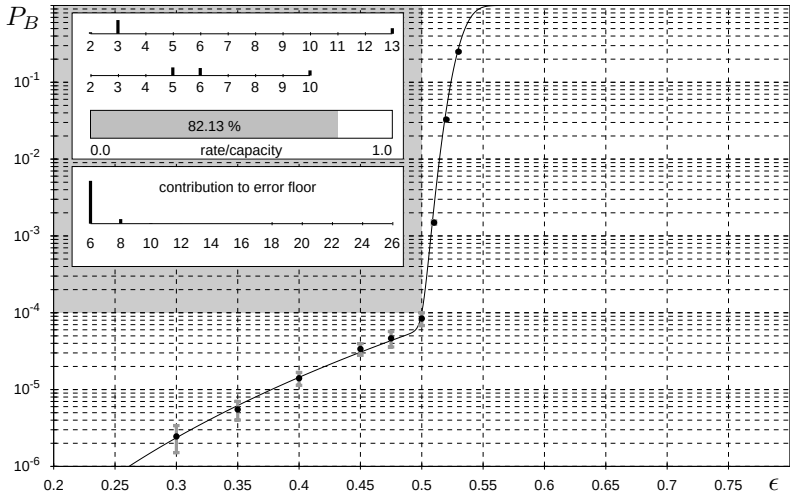


Figure 4.10: Error probability curve for the result of the optimization. The plain curve is $\mathrm{P}_B(\lambda, \rho)$ while the small dots are the simulation points with 95% confidence intervals. The degree distributions are $\lambda(x) = 0.0739196x + 0.657891x^2 + 0.268189x^{12}$ and $\rho(x) = 0.390753x^4 + 0.361589x^5 + 0.247658x^9$.

The block error probability plot for the result of the optimization is shown in Fig 4.10. Finally, as all our approach relied on using an approximation for the block error probability curves, we also show in Fig 4.10 simulation points that confirm that the result is as predicted.

Bibliography

[1] R. G. Gallager. *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, Massachusetts, 1963.

[2] V. Zyablov and M. Pinsker. Estimation of the error-correction complexity of Gallager low-density codes. *Problemy Peredachi Informatsii*, 11:23–26, Jan-Mar 1975.

[3] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27(5):533–547, September 1981.

[4] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, S-581 83, Linköping, Sweden, 1996.

[5] D. J. C. MacKay. Good error correcting codes based on very sparse matrices. *IEEE Trans. Inform. Theory*, 45:399–431, may 1999.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding. In *Proceedings of ICC'93*, pages 1064–1070, Geneve, Switzerland, May 1993.

[7] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proceedings of the 29th annual ACM Symposium on Theory of Computing*, pages 150–159, 1997.

[8] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman. Analysis of low density codes and improved designs using irregular graphs. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 249–258, 1998.

[9] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Trans. Inform. Theory*, 47(2):569–584, February 2001.

[10] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE Trans. Inform. Theory*, 47(2):585–598, 2001.

[11] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, February 2001.

[12] S.-Y. Chung, Jr. Forney, G. D., T. Richardson, and R. Urbanke. On the design of low-density parity-check codes within 0.0045 db of the Shannon limit. *IEEE Communications Letters*, 5(2):58–60, February 2001.

[13] T. Richardson, A. Shokrollahi, and R. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Trans. Inform. Theory*, 47(2):619–637, February 2001.

[14] N. Wiberg, H.-A. Loeliger, and R. Kötter. Codes and iterative decoding on general graphs. *European Transactions on Telecommuncations*, 6:513–526, September 1995.

[15] F. R. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, 2001.

[16] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke. Finite length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inform. Theory*, 48:1570–1579, June 2002.

[17] A. Montanari. Finite-size scaling of good codes. In *Proc. 39th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2001.

[18] T. Richardson and R. Urbanke. Efficient encoding of low-density parity-check codes. *IEEE Trans. Inform. Theory*, 47(2):638–656, February 2001.

[19] A. Orlitsky, K. Viswanathan, and J. Zhang. Stopping set distribution of ldpc code ensembles. *IEEE Trans. Inform. Theory*, 51(3):929–953, mar 2004. submitted.

[20] T. Richardson and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2006. In preparation.

[21] M. E. Fisher. Proceedings of the enrico fermi school, varenna, italy, 1970, course n. 51. In *Critical Phenomena*. International School of Physics Enrico Fermi, Course LI, edited by M. S. Green, (Academic, New York, 1971), 1971.

[22] V. Privman. *Finite Size Scaling and Numerical Simulation of Statistical Systems.* (World Scientific, Singapour, 1990).

[23] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke. Finite-length scaling for iteratively decoded ldpc ensembles. submitted to IEEE IT, June 2004.

[24] M. I. Friedlin and A. D. Wentzell. *Random Perturbations of Dynamical Systems.* Springer-Verlag, New York, 1984.

[25] C. McDiarmid. Concentration. In M. Habid, C. McDiarmid, R. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, number 16 in Algorithms and Combinatorics, pages 195–248. Springer, Berlin, 1998.

[26] S. R. S. Varadhan. *Lecure Notes on Stochastic Processes.* 2000. Available at `http://www.math.nyu.edu/faculty/varadhan/`.

[27] P. Groeneboom. Brownian motion with a parabolic drift and airy functions. *Probab. Th. Rel. Fields*, 81:79–109, 1989.

[28] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions.* Nat. Bur. Stand. 55, Washington, 1964.

[29] T. Richardson, A. Shokrollahi, and R. Urbanke. Finite-length analysis of various low-density parity-check ensembles for the binary erasure channel. In *IEEE International Symposium on Information Theory*, page 1, Lausanne, Switzerland, June 30–July 5 2002.

[30] B. Bollabas. *Random Graphs.* Cambridge University Press, 2001.

[31] H. S. Wilf. *Generatingfunctionology.* Academic Press, 2 edition, 1994.

[32] V. E. Britikov. The asymptotic number of forests from unrooted trees. *Math. Notes*, 43:387–394, 1988.

[33] V. F. Kolchin. *Random Graphs.* Cambridge University Press, 1999.

[34] W. Hoeffding. Probability inequalitites for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[35] K. Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19:357–367, 1967.

[36] A. Amraoui and R. Urbanke. Ldpcopt, a fast and accurate degree distribution optimizer for ldpc code ensembles. http://lthcwww.epfl.ch/research/ldpcopt, 2001.

[37] S. ten Brink. Convergence of iterative decoding. *Electron. Lett.*, 35(10):806–808, May 1999.

[38] A. Ashikhmin, G. Kramer, and S. ten Brink. Extrinsic information transfer functions: model and erasure channel property. *IEEE Trans. Inform. Theory*, 50(11):2657–2673, November 2004.

[39] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, New York, 1991.

[40] S.-Y. Chung. *On the construction of some capacity-approaching coding schemes.* PhD thesis, MIT, Cambridge, Massachusetts, 2000.

[41] A. Amraoui, S. Dusad, and R. Urbanke. Achieving general points in the 2-user Gaussian MAC without time-sharing or rate-splitting by means of iterative coding. In *IEEE International Symposium on Information Theory*, page 334, Lausanne, Switzerland, June 30–July 5 2002.

[42] A. Grant, B. Rimoldi, R. Urbanke, and P. Whiting. Rate-splitting multiple access for discrete memoryless channel. *IEEE Trans. Inform. Theory*, IT–47(3):873–890, March 2001.

Curriculum Vitae

Abdelaziz Amraoui

Address
Route Panoramique de la Corniche,
7000 Bizerte, Tunisia
Phone : +216 72 434702
e-mail : abdelaziz.amraoui@gmail.com

Status
Born 1^{st} June 1978
in Ankara, Turkey.
Citizen of Tunisia.

EDUCATION

2001-2006 **École Polytechnique Fédérale de Lausanne (EPFL), Switzerland**
Ph.D. thesis: *"Asymptotic and Finite-Length Optimization of LDPC Codes"*,
under the supervision of Prof. Rüdiger Urbanke.

2000-2001 **Pre-Doctoral School at EPFL, Lausanne, Switzerland**
School of computer and communications sciences,
Project: *Optimization of LDPC Codes Degree Distributions*,
under the supervision of Prof. Rüdiger Urbanke.

1997-2000 **École Nationale de l'Aviation Civile, Toulouse, France**
Electrical engineering degree and master of science in digital communication systems.

1995-1997 **Institut préparatoire aux études scientifiques et techniques, La Marsa, Tunisia**
"Mathématiques superieures et spéciales".

1990-1995 **Lycée Louis Massignon, Abu Dhabi, United Arab Emirates**
"Baccalauréat scientifique".

EMPLOYMENT HISTORY

08/2002–11/2002 **Summer intern** at the mathematics of communications research department at Bell Laboratories, Lucent Technologies, in Murray Hill, New Jersey.
Research area: Coding for broadcast channels.

09/2001–03/2006 **Research and teaching assistant** at the communication theory laboratory EPFL, Lausanne, Switzerland.
Research area: Analysis of iterative coding.

02/2000–08/2000 **Internship** at SITA, "Société Internationale de Télécommunications Aéronautiques" at Geneva headquarters.
Subject: Integration of a Voice-over-IP service to the IEEE-802.11 wireless LAN offering for the civil aviation community.

TEACHING EXPERIENCE

2001–2006 **Teaching assistant** at the communication theory laboratory EPFL, Lausanne, Switzerland.

Introduction to communications systems
School of computer and communication sciences, first year course.

Principles of digital communications
School of computer and communication sciences, third year.

Advanced digital communications
School of computer and communication sciences, fourth year.

Modern coding theory
School of computer and communication sciences, doctoral school.

HONORS AND AWARDS

2000	**Graduated ranked first**, electrical engineering École Nationale de l'Aviation Civile, Toulouse, France.
2000	**Prix de la Ville de Toulouse**, prize for best project (internship at SITA) École Nationale de l'Aviation Civile, Toulouse, France.
1997-2000	**Fellowship** from the French Ministry of Foreign Affairs. École Nationale de l'Aviation Civile, Toulouse, France.

LANGUAGE SKILLS

English, fluent

French, native

Arabic, native

Spanish, basic

PUBLICATIONS

Journals

[1] A. Amraoui, A. Montanari, T. Richardson and R. Urbanke,
Finite-Length Scaling for Iteratively Decoded LDPC Ensembles,
submitted to IEEE Trans. on Information Theory, June 2004

[2] A. Amraoui, A. Montanari, T. Richardson and R. Urbanke,
Finite-Length Scaling of Irregular LDPC ensembles,
to be submitted to IEEE Trans. on Information Theory, 2006

Conferences

[1] A. Amraoui, S. Dusad and R. Urbanke,
Achieving General Points in the 2-user Gaussian Mac without Time-Sharing or Rate-Splitting by Means of Iterative Coding, in Proc. IEEE International Symposium on Information Theory, Lausanne, Switzerland, June 30-July 5 2002, p. 334.

[2] A. Amraoui, G. Kramer and S. Shamai,
Coding for the MIMO Broadcast Channel, in Proc. IEEE International Symposium on Information Theory, Yokohama, Japan, June 29-July 4, 2003, p. 296.

[3] A. Amraoui, A. Montanari, T. Richardson and R. Urbanke,
Finite-Length Scaling for Iteratively Decoded LDPC Ensembles, in Proc. 41th Annual Allerton Conference on Communication, Control and Computing, Monticello, Illinois, October 2003.

[4] A. Amraoui, A. Montanari, T. Richardson and R. Urbanke,
Further Analysis of Finite-Length Scaling for Iteratively Decoded LDPC Ensembles, in Proc. IEEE International Symposium on Information Theory, Chicago, Illinois, June 2004.

[5] A. Amraoui, A. Montanari, T. Richardson and R. Urbanke,
Finite-Length Scaling and Finite-Length Shift for Low-Density Parity-Check Codes, in Proc. 42th Annual Allerton Conference on Communication, Control and Computing, Monticello, Illinois, Sept-Oct 2004.

[6] A. Amraoui, A. Montanari and R. Urbanke,
Finite-Length Scaling of Irregular LDPC Code Ensembles, in Proc. IEEE Information Theory Workshop, Rotorua, New-Zealand, Aug-Sept 2005.

[7] A. Amraoui, A. Montanari and R. Urbanke,
How to Find Good Finite-Length Codes: From Art Towards Science, in Proc. 4^{th} International Symposium on Turbo Codes and Related Topics, Munich, 3-7 Apr 2006.

[8] A. Amraoui, A. Montanari and R. Urbanke,
Analytic Determination of Scaling Parameters, accepted in IEEE International Symposium on Information Theory, Seattle, 9-14 July 2006.